# A MIMIC-STRUCTURED REGRESSOR ANALYSIS

Here we provide an initial analysis of our mimic regressor under simplifying assumptions as explained below. (Our experimental results show that intuitions hold in real-world scenarios.)

**Proposition 3.1** (Risk Upper Bound, Informal)**.** *Given regularity conditions, we show our proposed mimic-structured has a risk upper-bound of*

$$\mathbb{E}\left[\left|\mathbb{E}_{p(y|x)}\left[r(y, M(x\odot b))\right] - \mathbb{E}_{\widehat{p}(y|x)}\left[r(y, \widehat{M}(x\odot b))\right]\right|\right] \leq c''\, t^{-\frac{1}{d+2}}$$

*where $t$ is the budgeted observations on the BDM and estimate $\widehat{M}$ is trained on an iid dataset $B = \{(x^{(j)}\odot b^{(j)}, \eta^{(j)})\}_{j=1}^{t}$, $\eta^{(j)} \sim M(x^{(j)}\odot b^{(j)})$.*

*Proof.* Under the following simplified binary setting where: 1) the BDM does not take in options and may be written as $M(x\odot b)$; 2) the BDM only outputs a probability of selecting $y = 1$, $M : \mathbb{R}^d \mapsto [0, 1]$; 3) the reward function is bounded by a constant, $|r(y, M(x\odot b))| \leq R$; 4) the reward function is Lipschitz smooth $|r(y, \eta) - r(y, \eta')| \leq L|\eta - \eta'|$, for $y \in \{0, 1\}$. Then, for all $x, b$, we have (using the triangle inequality):

$$\begin{aligned}
&\left|\mathbb{E}_{p(y|x)}\left[r(y, M(x\odot b))\right] - \mathbb{E}_{\hat{p}(y|x)}\left[r(y, \hat{M}(x\odot b))\right]\right| \\
&\leq \left|p(y=1\mid x)r(1, M(x\odot b)) - \hat{p}(y=1\mid x)r(1, \hat{M}(x\odot b))\right| \\
&\quad + \left|p(y=0\mid x)r(0, M(x\odot b)) - \hat{p}(y=0\mid x)r(0, \hat{M}(x\odot b))\right| \\
&\leq \left|p(y=1\mid x)r(1, M(x\odot b)) - \left(\hat{p}(y=1\mid x) + p(y=1\mid x) - p(y=1\mid x)\right) r(1, \hat{M}(x\odot b))\right| \\
&\quad + \left|p(y=0\mid x)r(0, M(x\odot b)) - \left(\hat{p}(y=0\mid x) + p(y=0\mid x) - p(y=0\mid x)\right) r(0, \hat{M}(x\odot b))\right| \\
&\leq \left|p(y=1\mid x)\big(r(1, M(x\odot b)) - r(1, \hat{M}(x\odot b))\big)\right| + \left|\big(p(y=1\mid x) - \hat{p}(y=1\mid x)\big)r(1, \hat{M}(x,b))\right| \\
&\quad + \left|p(y=0\mid x)\big(r(0, M(x\odot b)) - r(0, \hat{M}(x\odot b))\big)\right| + |(p(y=0\mid x) - \hat{p}(y=0\mid x))r(0, \hat{M}(x\cdot b))| \\
&\leq 2L\left|M(x\odot b) - \hat{M}(x\odot b)\right| + \Big(|p(y=1|x) - \hat{p}(y=1\mid x)| + |p(y=0\mid x) - \hat{p}(y=0\mid x)|\Big)R.
\end{aligned}$$

Then, further supposing: 1) $X$ is bounded a.s.; 2) $p$ is Lipschitz smooth, $|p(y\mid x) - p(y\mid z)| \leq C\|x - z\|$ for all $y, x, z$, and following Gyorfi & Kohler, 2007, we may bound

$$\mathbb{E}\left[|p(y=1|x) - \hat{p}(y=1\mid x)| + |p(y=0\mid x) - \hat{p}(y=0\mid x)|\right] \leq c\, n^{-\frac{1}{d+2}},$$

for a constant $c$ using a non-parametric partitioning estimate for $\hat{p}$ on an *iid* dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ (see Thm. 2 (Gyorfi & Kohler, 2007)). Similarly (with accompanying Lipschitz assumptions on $M$),

$$\mathbb{E}\left[|M(x\odot b) - \hat{M}(x\odot b)|\right] \leq c'\, t^{-\frac{1}{d+2}},$$

by using a respective non-parametric partitioning estimates for $\hat{M}$ on an *iid* dataset $B = \{(x^{(j)}\odot b^{(j)}, \eta^{(j)})\}_{j=1}^{t}$, $\eta^{(j)} \sim M(x^{(j)}\odot b^{(j)})$. Taking $t \in \mathcal{O}(n)$, we may then simplify:

$$\mathbb{E}\left[\left|\mathbb{E}_{p(y|x)}\left[r(y, M(x\odot b))\right] - \mathbb{E}_{\hat{p}(y|x)}\left[r(y, \hat{M}(x\odot b))\right]\right|\right] \leq c''\, t^{-\frac{1}{d+2}}.$$

□

We see that the error (on randomly collected data) then is dominated by the number of budgeted observations on the BDM, $t$. Furthermore, one may compare to a standard regression approach to estimate rewards, which must operate in the joint $2\cdot d$ space, $\hat{f}(x, b) \approx \mathbb{E}_{p(y|x)}\left[r(y, M(x\odot b))\right]$, where corresponding mean absolute error bounds for partition estimators yield a slower rate of $t^{-\frac{1}{2d+2}}$ (see Chapter 4.4 (Györfi et al., 2006)). Alternatively, viewing the regression task as approximating $2^d$ separate regressors in $d$ dimensional space $\hat{f}_b(x) \approx \mathbb{E}_{p(y|x)}\left[r(y, M(x\odot b))\right]$ would place an exponential (in $d$, data dimensionality) factor on the required number of budgeted acquisitions.

# B DATASET

Statistics of the dataset used in our experiments are shown in Table 1. The skin segmentation, statlog (shuttle), SUSY, sensorless drive, california housing datasets are used in biased synthetic expert, multi-expert, and interpretable settings, whereas pima diabetes and california housing datasets are used in the large language model setting.

Table 1: Dataset information

| **Dataset** | **Instances** | **Features** |
|---|---|---|
| Skin Segmentation | 245057 | 3 |
| Statlog (Shuttle) | 58000 | 7 |
| SUSY | 5000000 | 18 |
| Sensorless Drive | 58509 | 48 |
| California Housing | 20640 | 8 |

# C BASELINES

## C.1 `Modiste-KNN`

Bhatt et al. (2023) assumed the reward to be bounded and discrete, specifically $r : \mathcal{X} \times \mathcal{A} \to \{0, 1\}$. To estimate rewards for any context-action pair, the original Modiste KNN variant queries nearest neighbors in the context space from the replay buffer with the same action as the given input; however, as action space size increases, many actions remain unobserved during training. The original Modiste KNN used the default value of 0.5 for unobserved actions, leveraging the bounded $[0, 1]$ reward range. Since our DISS environment has unbounded continuous rewards, we adapt this approach by setting the estimated reward for unobserved actions to the mean reward observed in the current replay buffer (other statistics such as quantiles could alternatively be used).

Since our DISS environment has unbounded continuous rewards, we adapt this approach by setting the estimated reward for unobserved actions to the minimum reward observed in the current replay buffer (other statistics like mean or quantiles could alternatively be used).

Crucially, this modification affects only the internal reward estimation mechanism used by Modiste-KNN for action selection. The actual rewards used for performance evaluation and comparison across all methods are the original environment rewards obtained when the selected actions are executed. Therefore, all baselines are evaluated against the same ground truth reward signal, ensuring fair experimental comparison despite the internal estimation adjustments.

## C.2 `Modiste-UKNN`

As action space size increases, the possibility of the reward estimator not being able to acquire sufficient interactions with the environment for each of the action in action space increases. The workaround proposed in Appx. C.1 only addresses the issue of which default value to use for unobserved actions but does not eliminate the fundamental need for default value imputation when exact action matches are unavailable.

To address this limitation, we note that DISS actions are likely related—for instance, an action $a$ selecting a similar set of features as another action $a'$ should yield similar rewards, with similarity determined by the degree of feature overlap between the two actions. The `Modiste-UKNN` leverages this intuition by performing k-nearest neighbors queries directly in the joint context-action space, thereby eliminating the need for default value imputation entirely.

Rather than requiring exact action matches as in `Modiste-UKNN`, we use the following distance metric:

$$\text{distance}(\langle x, a \rangle, \langle x', a' \rangle) = \|x, x'\| + \nu \|a, a'\| \tag{C1}$$

where $\nu$ is a hyperparameter. This approach can always find similar context-action pairs from the replay buffer, even when the exact action has never been observed, thus using actual observed rewards rather than imputed default values for reward estimation.

### C.3 SpannerIGW

SpannerIGW natively operates over bounded zero-one rewards (Zhu et al., 2022). However, our DISS environment produces unbounded continuous rewards, making the algorithm theoretically inapplicable without modification. To enable SpannerIGW to function in our setting, we implement a reward normalization scheme during the learning phase only. We first collect rewards using randomly sampled context-action pairs to compute the $10^{\text{th}}$ and $90^{\text{th}}$ percentiles of the reward distribution. During SpannerIGW's training phase, we clamp rewards to these percentiles and linearly rescale them to [0,1] before feeding them to the SpannerIGW learning algorithm. This normalization is purely a technical preprocessing step to satisfy the algorithm's bounded reward assumption.

Critically, all performance evaluation and reported metrics use the original unbounded rewards from the DISS environment at test time. After SpannerIGW selects an action, we compute and report the true unbounded reward for that action. The reward normalization affects only the internal learning mechanism of SpannerIGW, not the evaluation of its performance. This approach ensures completely fair comparison with other baselines, as SpannerIGW receives no advantage from the reward modification. The modification simply allows the algorithm to operate in our unbounded reward setting while preserving identical evaluation criteria across all methods. The initially collected random samples are used solely for computing normalization parameters and are never used for policy training.

## D Timing

In Table 2, we compare and contrast the amount of time in seconds needed for our proposed bootstrap mimic regressor versus bootstrap xgb regressor to suggest a query under different acquisition function.

Table 2: Timing summary

| **Num. of Obs.** | | 500 | 1000 | 2000 | 4000 | 8000 |
|---|---|---|---|---|---|---|
| **Acq. Func.** | **Est.** | | | | | |
| MTSPM | XGB | $0.072 \pm 0.024$ | $0.143 \pm 0.054$ | $0.116 \pm 0.000$ | $0.16985 \pm 0.026$ | $0.148 \pm 0.027$ |
| | Mimic | $0.071 \pm 0.016$ | $0.079 \pm 0.009$ | $0.134 \pm 0.004$ | $0.32505 \pm 0.179$ | $0.117 \pm 0.004$ |
| TS | XGB | $0.054 \pm 0.004$ | $0.099 \pm 0.013$ | $0.108 \pm 0.022$ | $0.09607 \pm 0.001$ | $0.159 \pm 0.012$ |
| | Mimic | $0.042 \pm 0.000$ | $0.088 \pm 0.000$ | $0.160 \pm 0.002$ | $0.28931 \pm 0.171$ | $0.197 \pm 0.086$ |
| Random | XGB | $0.067 \pm 0.013$ | $0.098 \pm 0.017$ | $0.147 \pm 0.020$ | $0.24118 \pm 0.144$ | $0.132 \pm 0.009$ |
| | Mimic | $0.045 \pm 0.001$ | $0.090 \pm 0.026$ | $0.093 \pm 0.003$ | $0.14160 \pm 0.006$ | $0.127 \pm 0.030$ |
| EI | XGB | $0.228 \pm 0.007$ | $0.253 \pm 0.022$ | $0.343 \pm 0.037$ | $0.59652 \pm 0.352$ | $0.491 \pm 0.082$ |
| | Mimic | $0.209 \pm 0.039$ | $0.272 \pm 0.136$ | $0.283 \pm 0.008$ | $0.50130 \pm 0.198$ | $0.352 \pm 0.062$ |
| REVI | XGB | $0.190 \pm 0.042$ | $0.207 \pm 0.039$ | $0.278 \pm 0.032$ | $0.35456 \pm 0.081$ | $0.348 \pm 0.037$ |
| | Mimic | $0.203 \pm 0.034$ | $0.274 \pm 0.068$ | $0.291 \pm 0.030$ | $0.65934 \pm 0.301$ | $0.432 \pm 0.083$ |

## E Experiment Details

We run our experiment on Lenovo ThinkStation P520 with Intel Xeon W-2295 with 512GB of RAM.

**Expert Queries** Each experiment used an initial (random) acquisition budget of $500$ expert queries and a total budget of $8,000$ queries.

**Reward Estimators** The XGBoost ensemble score estimators consist of 2 XGBoost regressors (both trained on the entire set of collected examples) with `n_estimators` set to 64. Our structured mimic score estimator uses ensembles consisting of XGBoost classifiers with `n_estimators=64` for both the expert mimic and the ground truth classifier components of the model. For completeness, we also included Gaussian Process based score estimator with constant mean and radial basis function kernel; the parameters for mean and covariance functions are updated using Adam optimizer with learning rate set to $10^{-3}$; the GP update subroutine is called after each query.

**Action Space** We limited the search space of possible actions for a given context to a maximum of 5000 options; for any setting where the full size of the action space $\|\{0,1\}^d \times \mathbb{O}\| > 5000$, we searched over a random sample of 5000 actions.

### E.1 Biased Synthetic Expert Setting

**Overload Bias Experiments** The expert's `bias_level` parameter was set to 1.0, with `min_temp` $= 1.0$ and `bias_mult` $= 5.0$. We show complete reward curve including all baselines is shown in Fig. 1.

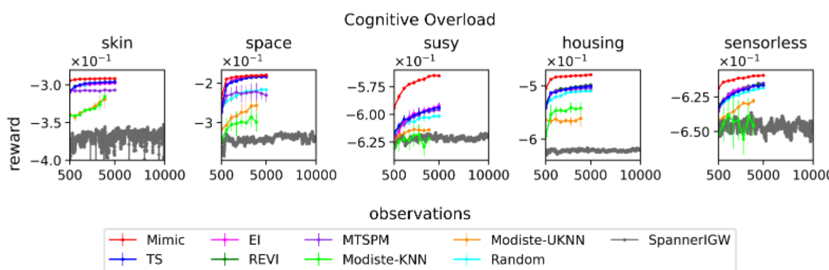



Figure 1: Average reward vs. observation data budget in the overload bias environments.

**Simplicity Bias Experiments** We set `bias_level=1.0`; the `poison_feature_index` was set independently for each dataset by training univariate classifiers for the ground truth data on each feature separately and selecting the feature that resulted in the best performance. We show complete reward curve including all baselines is shown in Fig. 2.

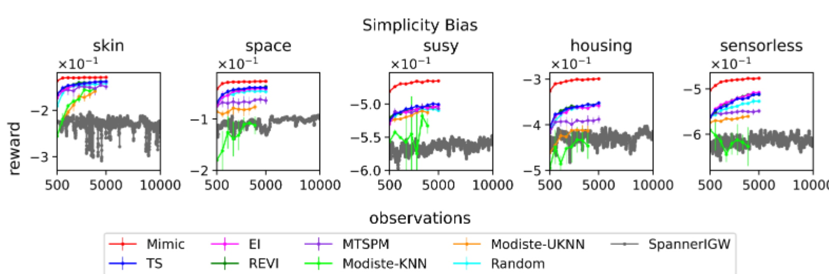



Figure 2: Average reward vs. observation data budget in the simplicity bias environments.

### E.2 Multi-Expert Setting

**Expert Configuration** As previously mentioned, we used K-Means clustering to simulate knowledge bases for different experts; for our experiments, we considered `k=4` experts and did not apply any added biases or limitations to the experts' decisions. We show complete reward curve including all baselines is shown in Fig. 3.

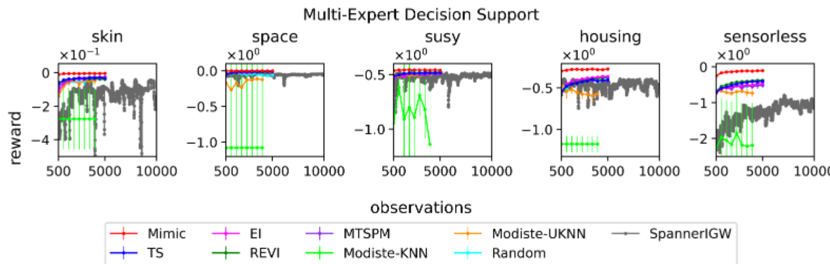



Figure 3: Average reward vs. observation data budget in the multi-expert environments.

### E.3 VISUALIZATION AND INTERPRETABILITY

We show complete reward curve including all baselines is shown in Fig. 4. We also show Accuracy and AUROC of `Mimic` selected features vs. random features and full feature classification (average selected reported) in Fig. 5. We note that number of features selected drastically decrease as lambda increases without degrading in task performance metrics. The phenomena are more significant in higher dimensional settings such as `space` and `susy`; on susy dataset, the policy on average select 8.30 features when $\lambda = 0$ compare to 1.65 when $\lambda = 0.9$; similar could also be observed for space dataset.

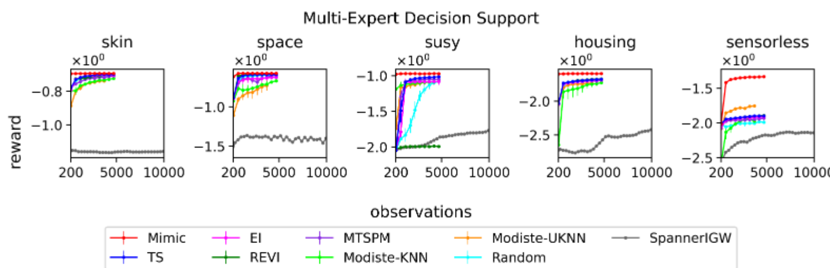
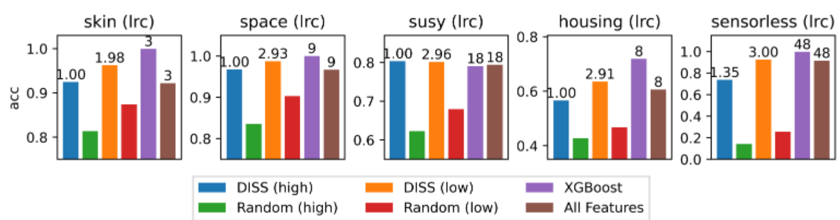
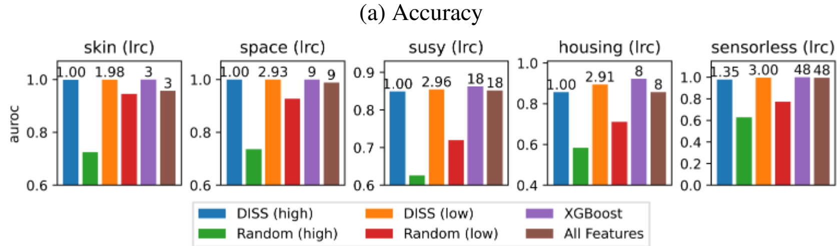



Figure 4: Average reward vs. observation data budget in the interpretability environments.




(a) Accuracy




(b) AUROC

Figure 5: Accuracy and AUROC of `Mimic` selected features vs. random features and full feature classification (average selected reported); low corresponds to $\lambda = 0.0$ and high corresponds to $\lambda = 0.9$. Numbers above the bar are the average number of features selected.

### E.4 Large Language Model Setting

The following are is an example contrasting the LLM outputs obtained using our `Mimic` approach vs `random` selection of features for the same instance. In the example for `Mimic` based subselection of features, notice the choice of salient features for the patient, enabling the LLM to generate better prediction, confidence and logical explanations. Compared to `random` selected features, wherethe LLM outputs fall short in prediction, confidence and explanations, leading to a misclassification (prediction of "Less than $250,000 for an opposite case). For all of the experiments, the factor that controls the randomness in output, `temperature`, was set to 0.7. For each input case, we extract the features using policies described in previous sections which are incorporated in the prompt, guiding the LLM to make predictions based on the selected features (selected on a per instance basis). The LLM returns a natural text as an output that specifies prediction, probability, and an explanation for this prediction. We parse the text to select the probability of this prediction to return a reward.

**Prompt for Mimic based selection of features**

You are an AI model trained to predict whether the median house price in a neighbourhood is less than $250,000 or greater than or equal to $250,000 based on certain features, specifically for the year 1990. Your task is to analyze the given features from this 1990 housing dataset and provide a prediction. Keep in mind that these prices are from over 30 years ago, so your analysis should consider the economic context of that time. **Some important notes:** Not all features may be provided for every prediction. Please interpret these data as you can and state the reasoning behind your prediction.
You will be given some or all of the following features for a neighbourhood, based on 1990 data: "Median income in block group (in thousands of dollars)", "Median house age in block group (in years)", "Average number of rooms per household", "Average number of bedrooms per household", "Block group population", "Average number of household members", "Block group latitude", "Block group longitude".

1. **Prediction:** State whether the median house price in 1990 is likely to be "Less than $250,000" or "Greater than or equal to $250,000". and mention how confident you are in your prediction [Low, Moderate, High].
2. **Confidence:** Express your confidence as a probability strictly in this format the scores should range from 0.0 to 1.0
3. **Provide explanation for your predictions:** Please provide your prediction for the following: 11.0 (Median house age in block group (in years)), 6.758558750152588 (Average number of rooms per household), 3.511711597442627 (Average number of household members), -122.23999786376953 (Block group longitude).

LLM output (for Mimic based selection of features)

**1. Prediction:** Positive, High 2. **Confidence:** 0.8 3. **Explanation:**
First, I'll consider the median house age. A median house age of 11 years suggests that the neighbourhood is relatively new, which might indicate higher house prices. However, without information on the original purchase price or construction costs, it's difficult to make a definitive conclusion.
Next, I'll examine the average number of rooms per household. With 6.76 rooms, this is a moderate number, which could suggest a mix of older and newer homes. This might not provide strong evidence for either a high or low house price.
The average number of household members is 3.51, which is slightly above the average US household size in 1990 (around 3.17 people per household). This could indicate a moderate level of household density, which might be associated with higher house prices.
The longitude of -122.24 places this neighbourhood in the San Francisco Bay Area, California, which was known for its high housing costs in 1990. This geographic location alone suggests that the median house price is likely to be greater than or equal to $250,000.
Considering the available data and the economic context, I'll make a prediction based on the available features and the strong influence of the location.

Prompt for random selection of features

You are an AI model trained to predict whether the median house price in a neighbourhood is less than $250,000 or greater than or equal to $250,000 based on certain features, specifically for the year 1990. Your task is to analyze the given features from this 1990 housing dataset and provide a prediction. Keep in mind that these prices are from over 30 years ago, so your analysis should consider the economic context of that time. **Some important notes:** Not all features may be provided for every prediction. Please interpret these data as you can and state the reasoning behind your prediction.
You will be given some or all of the following features for a neighbourhood, based on 1990 data: "Median income in block group (in thousands of dollars)", "Median house age in block group (in years)", "Average number of rooms per household", "Average number of bedrooms per household", "Block group population", "Average number of household members", "Block group latitude", "Block group longitude".

1. **Prediction:** State whether the median house price in 1990 is likely to be "Less than $250,000" or "Greater than or equal to $250,000". and mention how confident you are in your prediction [Low, Moderate, High].
2. **Confidence:** Express your confidence as a probability strictly in this format the scores should range from 0.0 to 1.0
3. **Provide explanation for your predictions:** Please provide your prediction for the following: AveBedrms: 6.758558750152588 (Average number of bedrooms per household) and Longitude: -122.31999969482422 (Block group longitude)

**LLM output (for random selection of features)**

1. **Prediction:** Negative, High 2. **Confidence:** 0.05 3. **Explanation:** First, I'll check if there's any crucial information missing. In this case, I notice that we're missing MedInc (Median income in block group), HouseAge (Median house age in block group), AveRooms (Average number of rooms per household), Population (Block group population), and AveOccup (Average number of household members). However, since we only have two features, I'll proceed with the available data.
Based on the provided features, I'll make a prediction based on the following logic:
1. Average number of bedrooms per household (AveBedrms) is moderate (6.758558750152588 ). In 1990, this might indicate a lower-income area, which could lead to lower house prices. 2. The longitude value (-122.31999969482422) suggests a location in the San Francisco Bay Area, California, which was known for high housing costs in 1990.

## F Additional Figures for LLM Experiments

See Fig. 6.

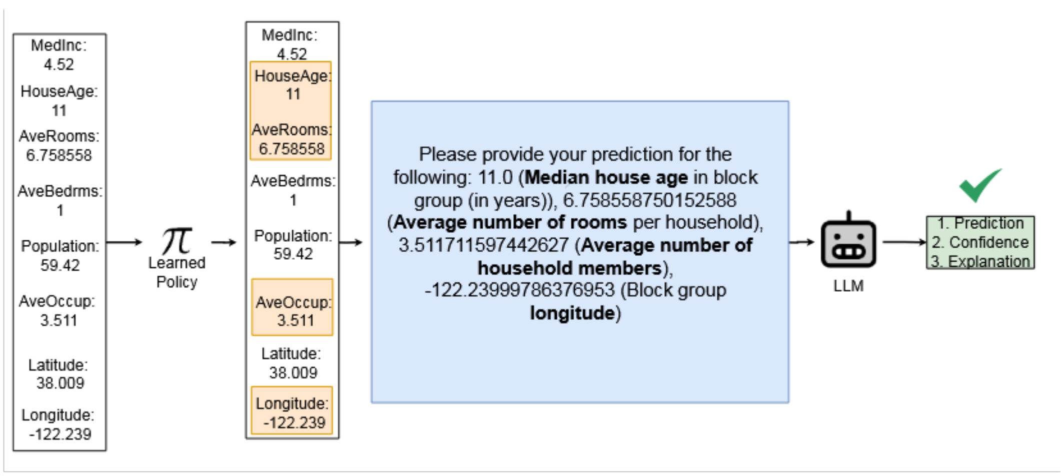



Figure 6: A graphical representation of our Mimic-based feature vs random feature selections(highlighted) sent to LLM as part of the prompt.

## G Sensitivity Analysis of Hyperparameters

In this section, we use the "simplicity bias" environment with mimic-structred regressor to conduct sensitivity analysis of various key hyperparameters used in our method, including initial capital $t_{\text{init}}$, number of available sub-sampled candidate $|A|$, and ensemble size $C$.

### G.1 Initial Capital $t_{\text{INIT}}$

Here, we show the effect of the number of warm-up interactions $t_{\text{init}}$ for each of the dataset. From Table 3, we found our method is relatively insensitive to the choice of $t_{\text{init}}$.

Table 3: Sensitivity analysis of initial capital

(a) skin

| Init Capital | Final Reward | Final Accuracy | Final AUROC | Final F1 |
|---|---|---|---|---|
| 10 | -0.1303 | 95.00 | 99.90 | 95.00 |
| 50 | -0.1300 | 94.90 | 99.87 | 94.90 |
| 125 | -0.1305 | 94.95 | 99.87 | 94.95 |
| 250 | -0.1290 | 94.95 | 99.89 | 94.95 |
| 500 | -0.1290 | 95.00 | 99.90 | 95.00 |

(b) space

| Init Capital | Final Reward | Final Accuracy | Final AUROC | Final F1 |
|---|---|---|---|---|
| 10 | -0.0430 | 99.80 | 100.000 | 99.800 |
| 50 | -0.0270 | 99.60 | 100.000 | 99.600 |
| 125 | -0.0306 | 99.30 | 100.000 | 99.300 |
| 250 | -0.0270 | 99.75 | 100.000 | 99.750 |
| 500 | -0.0278 | 99.75 | 100.000 | 99.750 |

(c) susy

| Init Capital | Final Reward | Final Accuracy | Final AUROC | Final F1 |
|---|---|---|---|---|
| 10 | -0.4696 | 78.65 | 85.15 | 78.65 |
| 50 | -0.4659 | 79.25 | 85.56 | 79.25 |
| 125 | -0.4639 | 79.15 | 85.76 | 79.15 |
| 250 | -0.4615 | 79.20 | 86.03 | 79.20 |
| 500 | -0.4628 | 78.85 | 85.56 | 78.85 |

(d) housing

| Init Capital | Final Reward | Final Accuracy | Final AUROC | Final F1 |
|---|---|---|---|---|
| 10 | -0.3024 | 87.90 | 92.74 | 87.90 |
| 50 | -0.3023 | 87.35 | 92.97 | 87.35 |
| 125 | -0.3038 | 87.85 | 92.50 | 87.85 |
| 250 | -0.3003 | 87.70 | 93.28 | 87.70 |
| 500 | -0.3003 | 87.75 | 92.94 | 87.75 |

(e) sensorless

| Init Capital | Final Reward | Final Accuracy | Final AUROC | Final F1 |
|---|---|---|---|---|
| 10 | -0.4800 | 95.70 | 98.47 | 95.70 |
| 50 | -0.4812 | 95.00 | 97.98 | 95.00 |
| 125 | -0.4744 | 94.85 | 98.19 | 94.85 |
| 250 | -0.4784 | 95.45 | 98.36 | 95.45 |
| 500 | -0.4786 | 95.75 | 98.18 | 95.75 |

### G.2 ENSEMBLE SIZE $C$

Here, we study the effect of $C$, the aforementioned ensemble size for posterior mean estimation, on reward along with other metrics. From the result shown in Table 4, we again see that our method is largely insensitive to ensemble sizes.

Table 4: Sensitivity analysis of ensemble size

(a) skin

| Metric | Size-2 | Size-5 | Size-10 | Size-30 |
|---|---|---|---|---|
| Reward | -0.13 | -0.13 | -0.129 | -0.13 |
| Accuracy | 95 | 95 | 95 | 95 |
| AUROC | 99.9 | 99.9 | 99.9 | 99.9 |
| F1-Score | 95 | 95 | 95 | 95 |

(b) space

| Metric | Size-2 | Size-5 | Size-10 | Size-30 |
|---|---|---|---|---|
| Reward | -0.0264 | -0.0297 | -0.027 | -0.0276 |
| Accuracy | 99.8 | 99.7 | 99.3 | 99.8 |
| AUROC | 100 | 100 | 100 | 100 |
| F1-Score | 99.8 | 99.7 | 99.3 | 99.8 |

(c) susy

| Metric | Size-2 | Size-5 | Size-10 | Size-30 |
|---|---|---|---|---|
| Reward | -0.467 | -0.469 | -0.466 | -0.468 |
| Accuracy | 79.3 | 78 | 79 | 78.8 |
| AUROC | 85.6 | 85.3 | 85.5 | 85.1 |
| F1-Score | 79.3 | 78 | 79 | 78.8 |

(d) housing

| Metric | Size-2 | Size-5 | Size-10 | Size-30 |
|---|---|---|---|---|
| Reward | -0.306 | -0.306 | -0.299 | -0.307 |
| Accuracy | 87.1 | 87.4 | 87.9 | 87 |
| AUROC | 92.9 | 92.6 | 93.1 | 92.6 |
| F1-Score | 87.1 | 87.4 | 87.9 | 87 |

(e) sensorless

| Metric | Size-2 | Size-5 | Size-10 | Size-30 |
|---|---|---|---|---|
| Reward | -0.483 | -0.478 | -0.473 | -0.473 |
| Accuracy | 95.6 | 95.2 | 96.1 | 96.2 |
| AUROC | 98.2 | 98.3 | 98.7 | 98.5 |
| F1-Score | 95.6 | 95.2 | 96.1 | 96.2 |

### G.3 NUMBER OF AVAILABLE CANDIDATES $|A|$

One fundamental combinatorial challenge is that the action space cardinality $|A| \geq 2^d$ grows exponentially with the number of features $d$, making exhaustive search intractable. Our approach addresses this through sampling a fixed number of candidate feature subsets at each decision point as described in Appx. E. Here, we conduct an ablation study to examine how the number of available candidates (1000, 2000, and 5000) affects the performance on high-dimensional datasets from the simplicity bias environment using Thompson sampling with mimic-structured regressor. From the result shown in Table 5, we see that including (searching over) a larger space of feature subsets only provides a marginal benefit past our choice of searching over 2000 random subsets, and results were largely stable to the hyperparameter of search space size. Note that the coarseness of a random search may also prevent overfitting to errors in the reward estimator. While other discrete optimization approaches may be applicable, the simplicity, lack of overhead, and good performance of our search heuristic make it a recommendable choice.

Table 5: Sensitivity analysis of available candidates

(a) susy

| Num. Cand. Avail. | Final Reward | Final Accuracy | Final AUROC | Final F1 |
|---:|---|---|---|---|
| 1000 | -0.4630 | 79.65 | 85.85 | 79.65 |
| 2000 | -0.4643 | 79.20 | 85.60 | 79.20 |
| 5000 | -0.4612 | 79.65 | 85.80 | 79.65 |

(b) sensorless

| Num. Cand. Avail. | Final Reward | Final Accuracy | Final AUROC | Final F1 |
|---:|---|---|---|---|
| 1000 | -0.4801 | 95.65 | 98.52 | 95.65 |
| 2000 | -0.4780 | 95.80 | 98.81 | 95.80 |
| 5000 | -0.4793 | 94.80 | 97.95 | 94.80 |

## H Complete Mimic-Structured Regressor Ablation

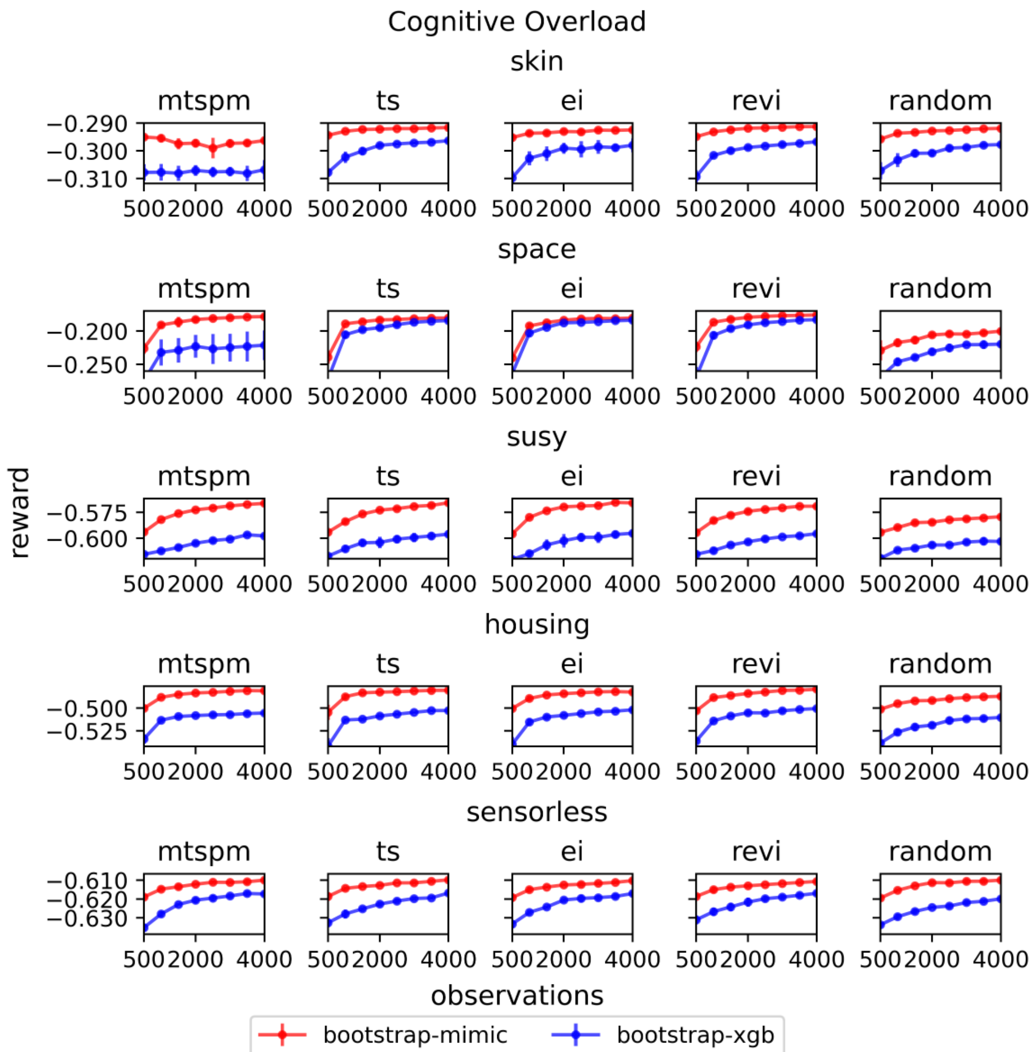



Figure 7: Cognitive overload environments

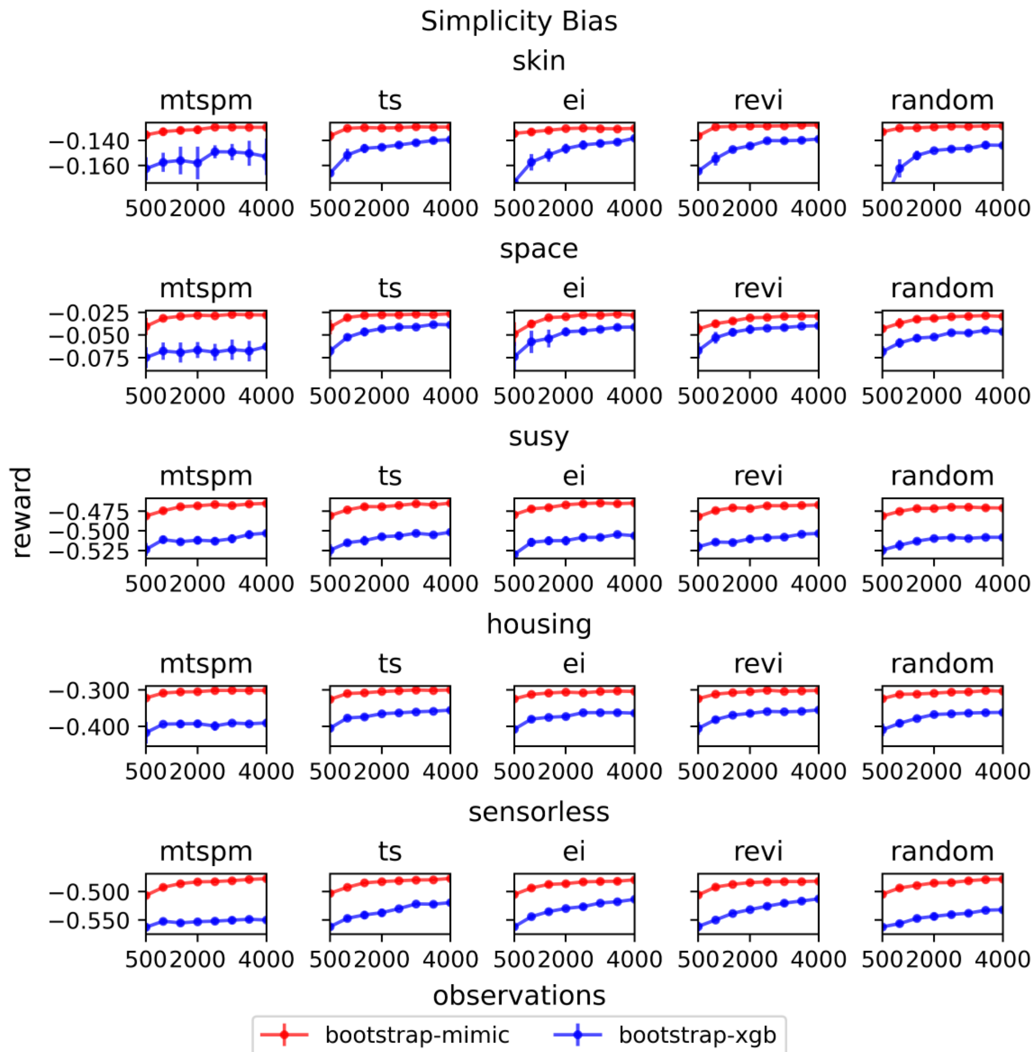



Figure 8: Siimpilcity bias environments

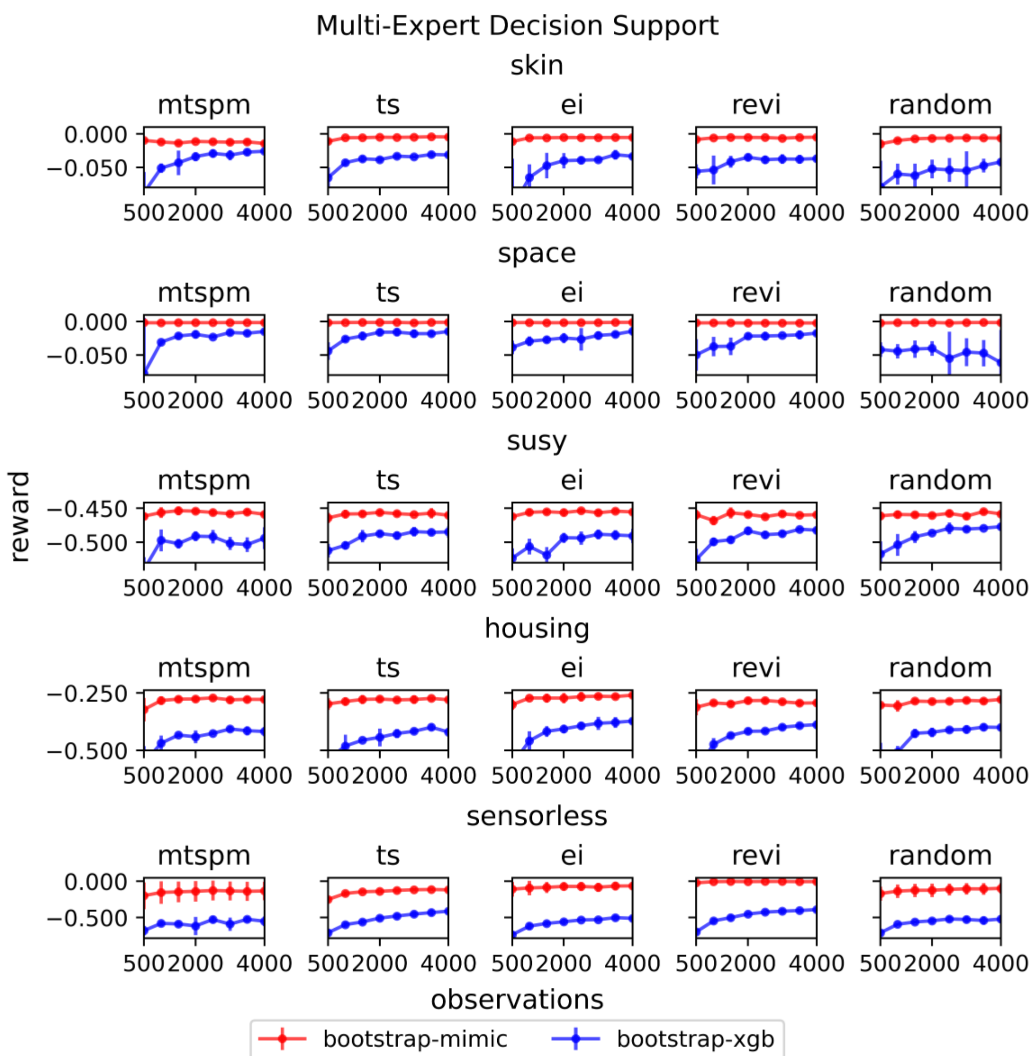



Figure 9: Multiple expert assignment environments

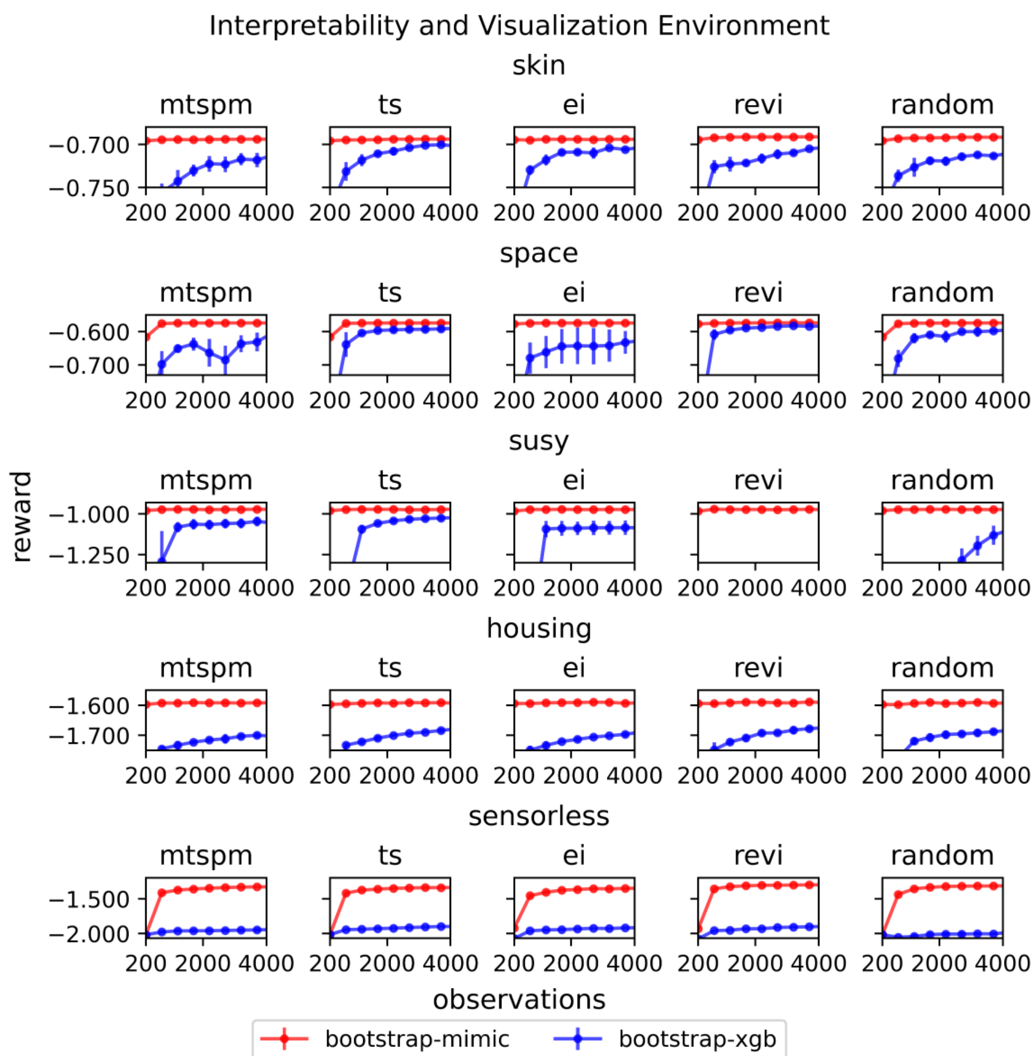



Figure 10: Inerpretability environments

### H.1 Further Discussion and Limitations

We present a novel Dynamic Information Sub-selection System that may be used to improve the downstream performance of existing models as well as potentially facilitate human decision-makers by presenting the most efficacious information at inference time. However, like any other ML system, it is important that one robustly test performance and limitations in sensitive applications, e.g. inadequate observation budget allocation may restrict the policy's predictive performance.

### H.2 (Optional) Additional Statements

#### Use of LLM

LLMs are used to aid or polish writings for all sections. Additionally, LLMs are used as the black-box decision maker in the experiment setting described in § 4.5.

## References

Umang Bhatt, Valerie Chen, Katherine M Collins, Parameswaran Kamalaruban, Emma Kallina, Adrian Weller & Ameet Talwalkar. Learning personalized decision support policiesin: *arXiv preprint arXiv:2304.06701*. 2023.

Lszl Gyorfi & Michael Kohler. Nonparametric estimation of conditional distributionsin: *IEEE Transactions on Information Theory* 53.5, pp. 1872–1879. IEEE, 2007.

László Györfi, Michael Kohler, Adam Krzyzak & Harro Walk. *A distribution-free theory of nonparametric regression*Springer Science & Business Media, 2006.

Yinglun Zhu, Dylan J Foster, John Langford & Paul Mineiro. Contextual Bandits with Large Action Spaces: Made Practicalin: *Proceedings of the 39th International Conference on Machine Learning*, pp. 27428–27453. PMLR, 17–23 Jul 2022.