A Appendix

A.1 User Study

We recruited 10 male, 1 female, and 1 non-binary participants, with an average age of 27. We obtained informed consent from each participant, as well as IRB approval for our study. Each participant was provided with the rules of the task, and a short practice period of 10 episodes to familiarize themselves with the 2D cursor control environment. Each participant was compensated with a $10 Amazon gift card.

When prompted to "please describe your command strategy", participants responded as follows.

> User 0:
> After Phase A:
> **Use 1-2 trials to get a sense of the angle difference**
> After Phase B:
> same as phase A
>
> User 1:
> After Phase A:
> **I positioned the cursor opposite of the triangle and the green dot since most of the physics made it such that that moved the triangle in the desired direction**
> After Phase B:
> same as phase A
>
> User 2:
> After Phase A:
> steering wheel
> After Phase B:
> same as phase A
>
> User 3:
> After Phase A:
> **Try to predict the angle of the offset, then course adjust**
> After Phase B:
> **Initially, I twirled my mouse around to find the correct heading, eventually I was able to predict the offset well enough to directly go to the goal position.**
>
> User 4:
> After Phase A:
> **I looked at where the circle went in the first few seconds and adjusted**
> After Phase B:
> Look at where the circle went in the first few seconds, then adjust the angle offset from there
>
> User 5:
> After Phase A:
> **I recalibrated as it started moving, it was easier to see what direction you're going in after you start moving**
> After Phase B:
> **If it moved in the direction of cursor, then I just clicked the green dot. If it was 180 rotation, I imagined just doing the opposite of what I normally did. If it was 90, it was the hardest.**
>
> User 6:
> After Phase A:
> **tried to identify the angle offset between curse and green dot so that it went straight to green dot**
> After Phase B:
> **Put the cursor where the green dot was**
>
> User 7:
> After Phase A:
> **try to figure out the angle over time**
> After Phase B:
> **again predicting the angle and adjusting on fly as need being. As algorithm seemed**

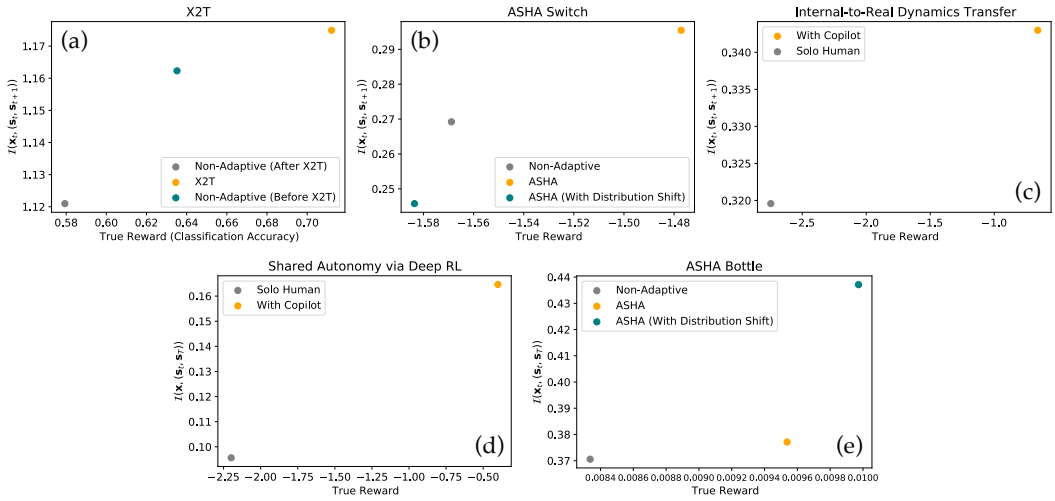

Figure 5: In each domain, the interface with the highest mutual information score is also the interface that enables users to achieve the highest ground-truth reward. Analogous to Fig. 2 in the main paper, but averaging the mutual information scores and ground-truth rewards across all 12 user study participants in each experimental condition.

to get better, began just hovering mouse over the target

User 8:
After Phase A:
I first tried to figure out what the constant angular offset was, and then tried to adjust my steering accordingly
After Phase B:
Pretty much the same as the first phase, trying to figure out the offset, and then steering accordingly

User 9:
After Phase A:
Try to learn the perturbation in the first couple episodes then adjust the cursor angle accordingly afterwards.
After Phase B:
Initially, my strategy was unchanged from Phase 1. After 30 episodes I did not perceive any perturbation between my cursor and the direction of the circle.

User 10:
After Phase A:
same as phase B
After Phase B:
Trying to determine the angle between the goal and where i should point my cursor, for each batch of 10

User 11:
After Phase A:
I tried to figure out the angle that was being added to my input and then adjusted my control to be minus that angle from the direction I actually wanted the dot to go. The angles that were close to either 0 or 180 were easier to control than those closer to 90/270 degrees.
After Phase B:
Sometimes I could figure out that there was some offset angle and then I would try to adjust my control based on that. Sometimes I couldn't tell what the transformation of my input was and then I was just making small adjustments until the dot headed in the right direction. Sometimes after a particularly easy block (when I just had to point at the green dot and it would go to it) I got unfocused and had a harder time controlling some of the more difficult blocks right after.

Table 1: Subjective Evaluations from User Study Participants

	p	Random Interfaces	MIMI
I felt in control	$> .05$	4.08	4.92
The system responded to my input in the way that I expected	$> .05$	4.17	4.75
The system improved over time	$< \mathbf{.01}$	3.33	**5.83**
I improved at using the system over time	$> .05$	5.58	6.08

Means reported for responses on a 7-point Likert scale, where 1 = Strongly Disagree, 4 = Neither Disagree nor Agree, and 7 = Strongly Agree. p-values from a one-way repeated measures ANOVA with the presence of MIMI as a factor influencing responses.

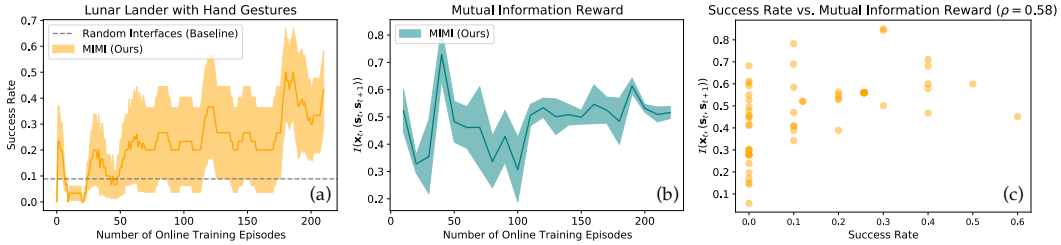

Figure 6: **(a)** From Fig. 4a. **(b)** Analogous to Fig. 3b, but for Lunar Lander instead of cursor control. **(c)** Analogous to Fig. 3c, but for Lunar Lander instead of cursor control.

To account for differences in the number of online training episodes for each of the 12 participants in Fig. 3a, we pad the sequence of true rewards with the final true reward, up to the maximum sequence length across participants. We use the same padding scheme to account for differences in the number of online training episodes for each of the 3 training runs in Fig. 4.

A.2 Implementation

In our cursor control and Lunar Lander user studies in Sections 5.2 and 5.3, we simplify the policy architecture to ignore the environment state $\mathbf{s}_t$, so that $\pi(\mathbf{a}_t|\mathbf{x}_t)$ is only conditioned on the user's command $\mathbf{x}_t$.

All user studies and experiments (including model training) were performed on a consumer-grade MacBook Pro laptop computer.

We use the `scikit-optimize` library[2] to implement the Bayesian optimization algorithm for mutual information maximization described in Sec. 3. In the cursor control user study, the first 5 interfaces are sampled uniformly at random, then the subsequent policies are selected using acquisition functions that measure expected improvement, lower confidence bound, and probability of improvement. In the Lunar Lander experiments, the first 3 interfaces are sampled uniformly at random. In both domains, we have the user operate each interface for 10 episodes.

Code and data are available at `https://github.com/rddy/mimi`.

A.3 Environments

The X2T experiments formulate typing as a contextual bandit problem. Hence, we format the dataset such that each attempt to press a button is a separate episode that consists of only one state transition, from the all-zeros initial state $\mathbf{s}_0$ (which represents no buttons being pressed) to a one-hot encoding $\mathbf{s}_1$ of the button that was pressed. In the ASHA experiments, the state $\mathbf{s}_t$ is either a 142-dimensional vector (switch domain) or 151-dimensional vector (bottle domain) that includes the 7 joint positions of the arm and the 2D position and 4D orientation of the end effector–the full list of state variables can be found in Sec. B.3 of the appendix in [64]. In our analysis of the SAvDRL and ISQL data, we set the state $\mathbf{s}_t$ to be the 1D horizontal position of the lander.

[2] `https://scikit-optimize.github.io/stable/auto_examples/bayesian-optimization.html`

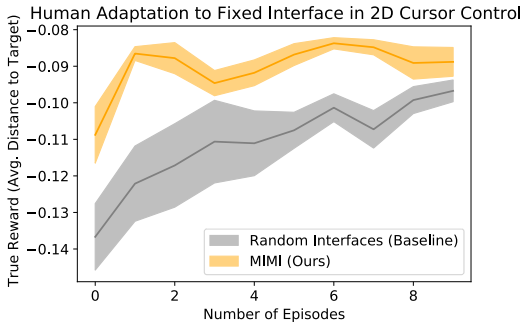

Figure 7: A view of the data from the user study in Fig. 3 that illustrates how a user adapts to a fixed interface. During the user study, the user is repeatedly presented with a new interface and asked to operate it for 10 episodes. During those 10 episodes, the user becomes more proficient at using that fixed interface. When presented with a random interface, the user takes a longer time to learn to use the interface and achieves worse final performance, compared to when they are presented with an interface that is being optimized by MIMI.

In X2T, ground-truth rewards correspond to the classification accuracy of the interface, which predicts the user's desired button (1 of 8) given the user's eye gaze signals. There were three conditions in the X2T experiments: the user first operated the non-adaptive baseline interface, then operated the adaptive X2T interface, then returned to operating the non-adaptive baseline interface. There were 12 participants in the X2T user study, yielding a total of $12 \cdot 3 = 36$ data points in the scatter plot in Fig. 2 (one for each user in each condition). In the ASHA switch experiments, ground-truth rewards penalize distance from the robot end effector to the position of the target switch. In the ISQL data, there were only two conditions: the user playing the Lunar Lander game on their own, and with assistance. In the Lunar Lander game, ground-truth rewards penalize crashing, and give a bonus for landing between the flags. In the ASHA bottle experiments, ground-truth rewards give a bonus for opening the door in front of the desired bottle and for reaching the desired bottle.

In the 2D cursor control environment, the ground-truth reward function is the average negative distance to target throughout the episode. The episode terminates when the user reaches the target, so naively computing this reward would penalize the user for reaching the target quickly. To address this issue, we treat reaching the target as entering an absorbing state, and re-weight the reward as $r' \leftarrow |\tau| \cdot r + (T - |\tau|) \cdot 0$, where r is the average negative distance to target throughout the trajectory τ, $|\tau|$ is the length of the trajectory, and T is the maximum episode length. The maximum episode length is $T = 300$ timesteps.

In the Lunar Lander experiments in Sec. 5.3, we set the state $\mathbf{s}_t$ to be the 3D position and orientation of the lander. The maximum episode length is 500 timesteps. The location of the landing zone is sampled uniformly at random at the beginning of each episode.