Simulating Errors in Touchscreen Typing (Supplementary Material)

A JOINT OPTIMIZATION OF PARAMETERS

We used a two-loop process to optimize Typoist, designed to find the best set of reward ($\boldsymbol{r} \in \mathcal{X}_r$), model ($\boldsymbol{\theta}_{model} \in \mathcal{X}_{model}$), and human parameters ($\boldsymbol{\theta}_{human} \in \mathcal{X}_{human}$). We aimed to minimize the Jensen-Shannon Divergence, $JS(.)$, between the typing patterns of the agent, $Metrics(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, ..., \boldsymbol{\tau}_n)$, and actual humans, $Metrics(h_1, h_2, ..., h_n)$, ensuring realistic and effective interactions.

Both loops are guided by a Bayesian Optimization (BO) framework, chosen for its efficiency in handling costly black-box objective functions. BO uses a probabilistic surrogate model to predict the performance of different parameter sets, reducing the number of evaluations required. This approach allows for an efficient search through the parameter space, resulting in a robust typing model that performs well across various scenarios.

The overall optimization algorithm is presented in Algorithm 1.

Algorithm 1 Joint Optimization of Parameters

Require: Define the search space of parameters to optimize.
$\mathcal{X}_r$: search space of reward parameters;
$\mathcal{X}_{model}$: search space of model parameters;
$\mathcal{X}_{human}$: search space of human parameters.

while *outer loop*: optimal $\boldsymbol{r}^*$ and $\boldsymbol{\theta}^*_{model}$ has not been found **do**
    Jointly select $\boldsymbol{r} \in \mathcal{X}_r$ and $\boldsymbol{\theta}_{model} \in \mathcal{X}_{model}$ using Bayesian optimization.
    Train model $\mathcal{M}(\boldsymbol{\theta}_{model}, \tilde{\boldsymbol{\theta}}_{human}, \boldsymbol{r})$ with $\tilde{\boldsymbol{\theta}}_{human} \in \mathcal{X}_{human}$ randomly sampled in each training episode.
    while *inner loop*: optimal $\boldsymbol{\theta}^*_{human}$ has not been found **do**
        Select parameters $\boldsymbol{\theta}_{human} \in \mathcal{X}_{human}$ via Bayesian optimization
        Sample n typing episodes $(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, ..., \boldsymbol{\tau}_n)$ from the model $\mathcal{M}(\boldsymbol{\theta}_{model}, \boldsymbol{\theta}_{human}, \boldsymbol{r})$
        Compute $Metrics(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, ..., \boldsymbol{\tau}_n)$
        Evaluate with human data: $JS(Metrics(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, ..., \boldsymbol{\tau}_n), Metrics(h_1, h_2, ..., h_n))$
    end while
end while
return: general model $\mathcal{M}(\boldsymbol{\theta}^*_{model}, \boldsymbol{\theta}^*_{human}, \boldsymbol{r}^*)$, and user-fitted parameters $\boldsymbol{\theta}^*_{human}$

A.1 Outer Loop Optimization

The outer loop focuses on optimizing parameters that influence both the reward system, and the model's parameters. Key parameters optimized in this loop include:

- **Reward Parameters ($\boldsymbol{r} \in \mathcal{X}_r$):** These parameters control how rewards are given to the RL agent based on typing accuracy and time efficiency. The total reward received by the agent is defined as:

$$\mathcal{R} = (1 - \mathrm{Err}^{\alpha}) - w \cdot t, \tag{1}$$

  where:
  - α: exponential power constant that adjusts the sensitivity of the reward to the error rate (Err).
  - w: weight for time in the reward calculation.
  - Err: error rate, indicating the percentage of incorrectly typed characters.

- t: typing time.

The reward parameters optimized in the outer loop are $\boldsymbol{r} = [w, \alpha]$.

- **Model Parameters ($\theta_{model} \in \mathcal{X}_{model}$):** These parameters control various aspects of the agent's training process, such as exploration and learning from experiences:
  - **Lambda parameter for the Generalized Advantage Estimation (GAE)**, which affects the trade-off between bias and variance in advantage estimation during training.
  - **Discount factor** for future rewards, which determines how much the agent values immediate rewards compared to future rewards.
  - **Learning rate**, which defines the step size for the optimization algorithm and affects how quickly or slowly the model learns from new data.
  - **Number of training epochs**, which determines how many times the learning algorithm updates the model parameters per training iteration.
  - **Entropy coefficient**, which encourages exploration by penalizing deterministic policies, helping the agent explore a wider range of actions.
  - **Clipping range for policy updates**, which helps stabilize training by preventing large updates that could destabilize the policy.
  - **Neural network architecture** define the architecture of the neural networks used by the RL agent, which affects its ability to learn and generalize from data. Specifically, the number of layers and the number of neurons in each layer of the policy network are optimized for the typing agent.

A.2 Inner Loop Optimization

Within each iteration of the outer loop, the inner loop optimizes human parameters of errors, $\boldsymbol{\theta}_{human} \in \mathcal{X}_{human}$, to adapt the agent to different user behaviors, which are composed of the following:

- **Parameter of proofreading errors:** Accounts for mistakes made during the proofreading process.
- **Parameter of guidance mistakes:** Measures errors due to incorrect guidance or decision-making while typing.
- **Parameter of forgetting rate:** Captures the rate at which the user forgets key placements or typed content.
- **Parameter of bouncing rate:** Represents the frequency of repeated key presses (e.g., double-pressing a key unintentionally).
- **Parameter of order errors:** Reflects the occurrence of typing characters in the wrong order.
- **Parameter of finger dexterity:** Represents the user's typing speed and finger movement efficiency.

This inner loop ensures that the model is adaptable to individual user behaviors, enhancing its generalization capabilities.

B BAYESIAN *T*-TEST

We conducted a statistical analysis using the Bayes Factor from a t-test function to evaluate how closely the simulated data from Typoist and CRTypist align with the human data distribution. For each model (Typoist and CRTypist), we simulate 30 data points, the same as the number of human data. We define the null hypothesis (H_0) as no difference between the simulated data and human data, while the alternative hypothesis (H_1) asserts that such a difference exists. The test results suggest that Typoist aligns more closely with human data than CRTypist across most metrics in both the Finnish typing dataset and the Gboard dataset. All BF_{10} values are shown in Table B1.

Table B1. Statistical analysis using Bayes Factors (BF10) from the *t*-test function, where H0 represents no difference between the simulated and human data, and H1 represents a difference.

Model	**User Group**	**Metric**	**BF10**	**Support**
Typoist	Finnish typists	WPM	0.576	H0
		Uncorrected error (%)	0.382	H0
		Corrected error (%)	0.302	H0
		KSPC	0.593	H0
		Backspaces	0.332	H0
		Immediate corrections	81.015	H1
		Delayed corrections	0.291	H0
	Gboard typists	WPM	0.324	H0
		Uncorrected error (%)	1.019	H1
		Corrected error (%)	0.266	H0
		KSPC	1.282	H1
		Backspaces	0.336	H0
		Immediate corrections	0.263	H0
		Delayed corrections	0.41	H0
CRTypist	Finnish typists	WPM	2.477	H1
		Uncorrected error (%)	0.262	H0
		Corrected error (%)	1.86	H1
		KSPC	1.985	H1
		Backspaces	1.184	H1
		Immediate corrections	17170.0	H1
		Delayed corrections	0.529	H0
	Gboard typists	WPM	$1.577 \cdot 10^{115}$	H1
		Uncorrected error (%)	$7.163 \cdot 10^{130}$	H1
		Corrected error (%)	$+\infty$	H1
		KSPC	$+\infty$	H1
		Backspaces	$2.536 \cdot 10^{98}$	H1
		Immediate corrections	$2.015 \cdot 10^{276}$	H1
		Delayed corrections	$2.534 \cdot 10^{59}$	H1