

ECE 209AS Fall 2020 Bake-off Demo, 10/28/20

---

# HMM-T9: An AI-enabled Tiny T9 Keyboard

---

**Viacheslav Inderiakin** and **Swapnil Sayan Saha**  
Dept. of ECE, UCLA  
Team: Orion

# Design Intuition

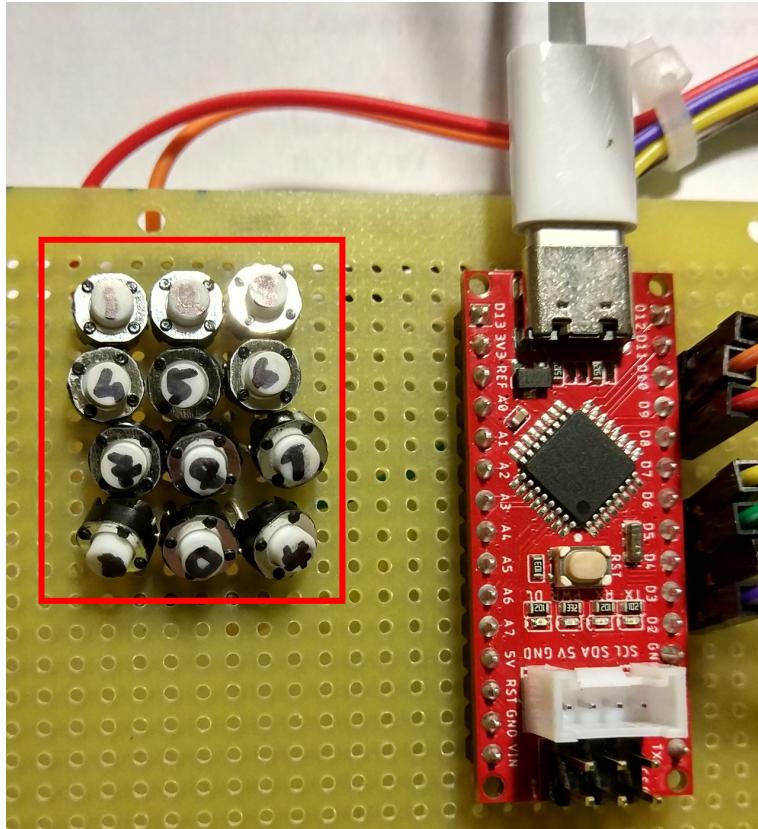
Natural and adopted layout

Strong feedback

Robust to elements and user changes

Touch-screen independent

AI-enabled

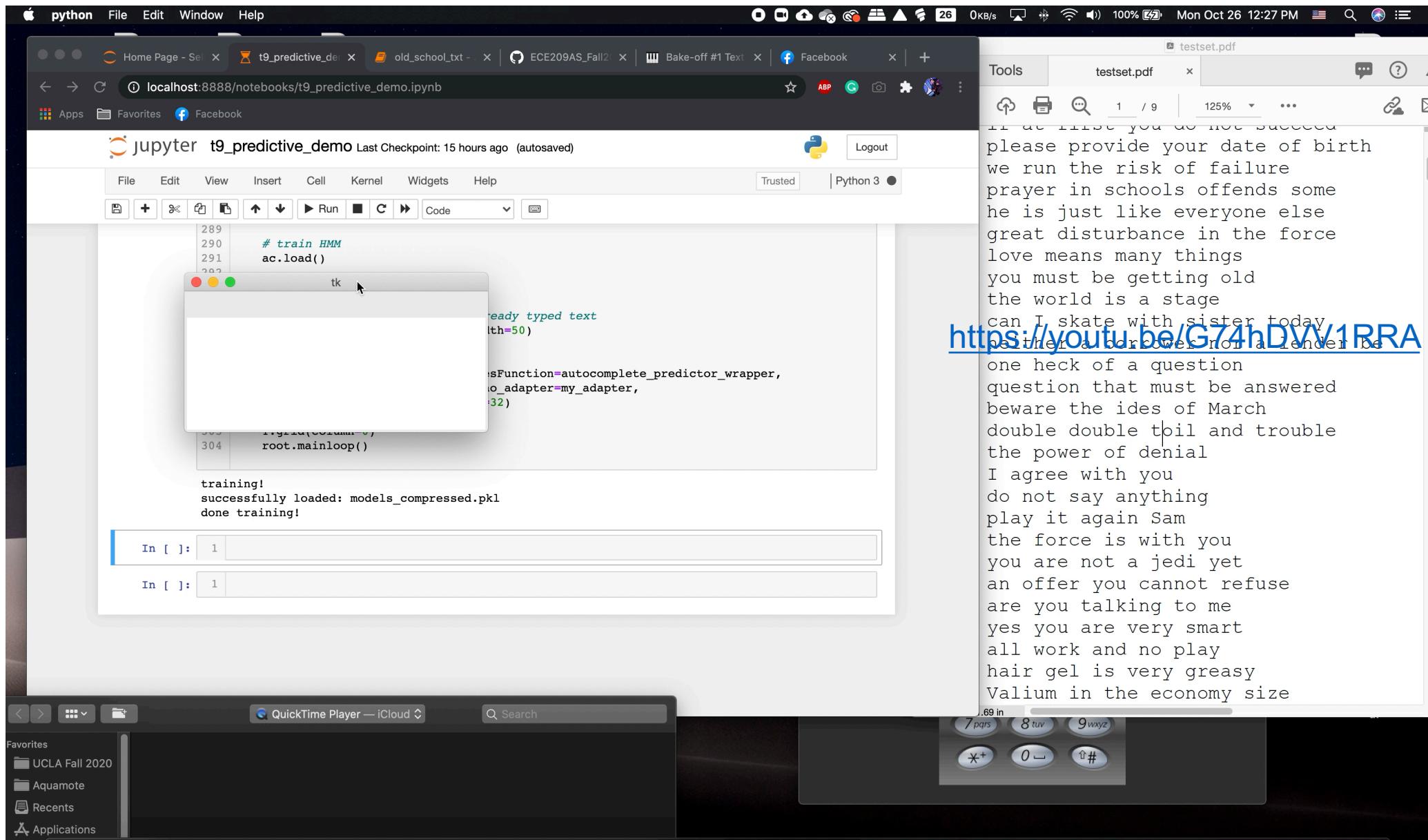


HMM-T9 PCB (fabricated at NESL, UCLA)

HMM-T9 (Veroboard version)

- HMM-T9 – A tiny T9 keyboard using Hidden Markov Models (HMM) for text prediction.
- Contact area: 2.95 cm<sup>2</sup> usable contact area, spread over ~ 2.2 X 1.7 cm (veroboard).
- USB-C communication for broad compatibility spectrum.
- Lightweight-AI assisting familiar layout.

# 1x Recording

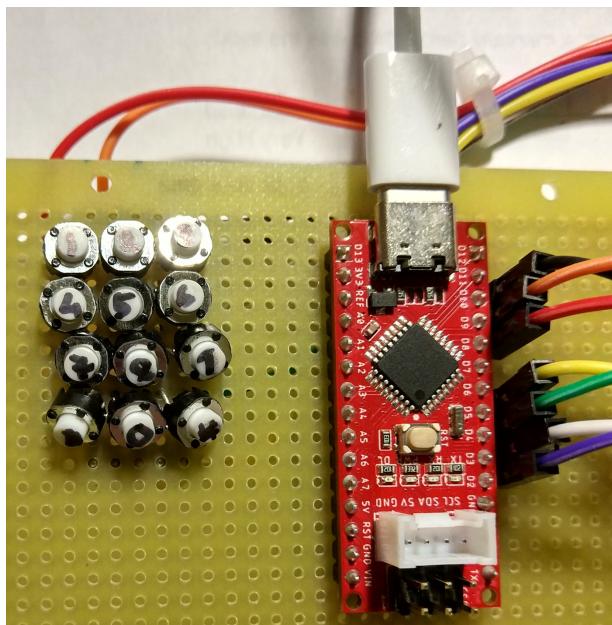


# Demo



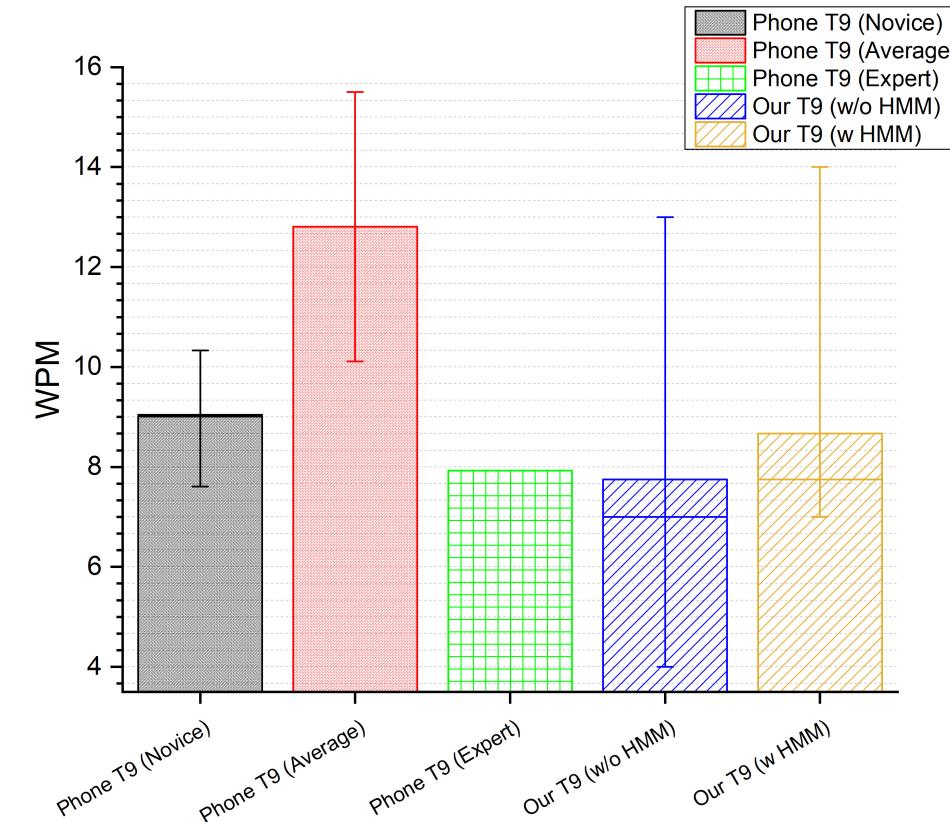
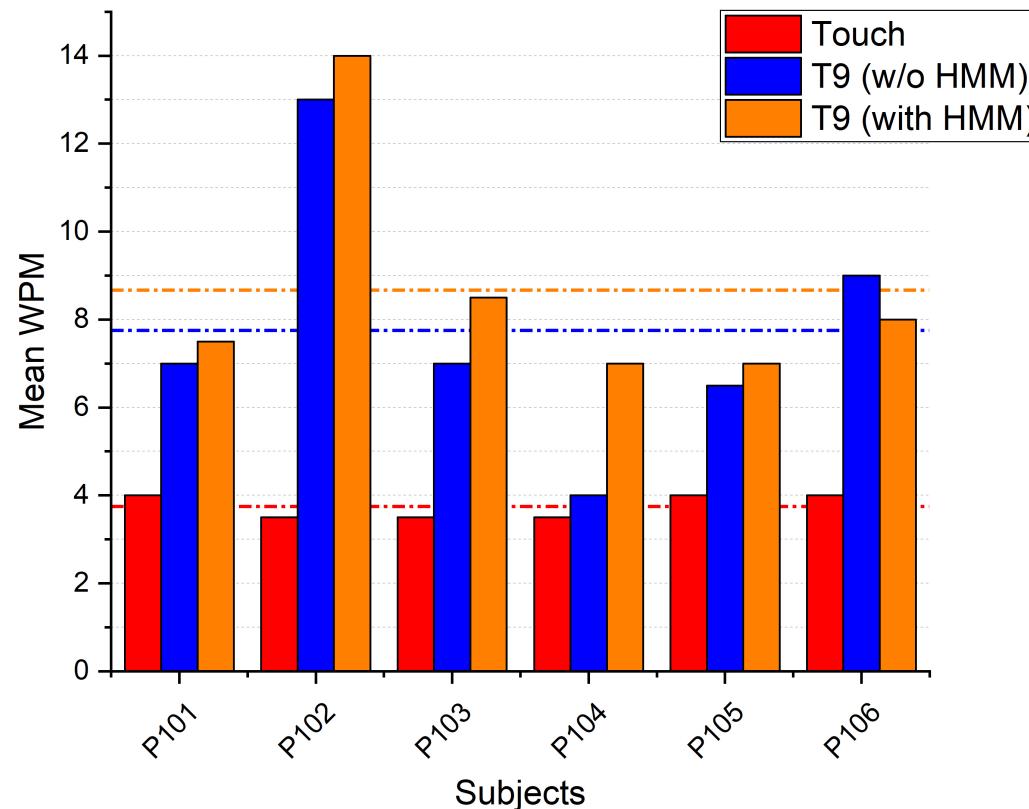
<https://youtu.be/AMEvziVtORE>

# Evaluation Criteria



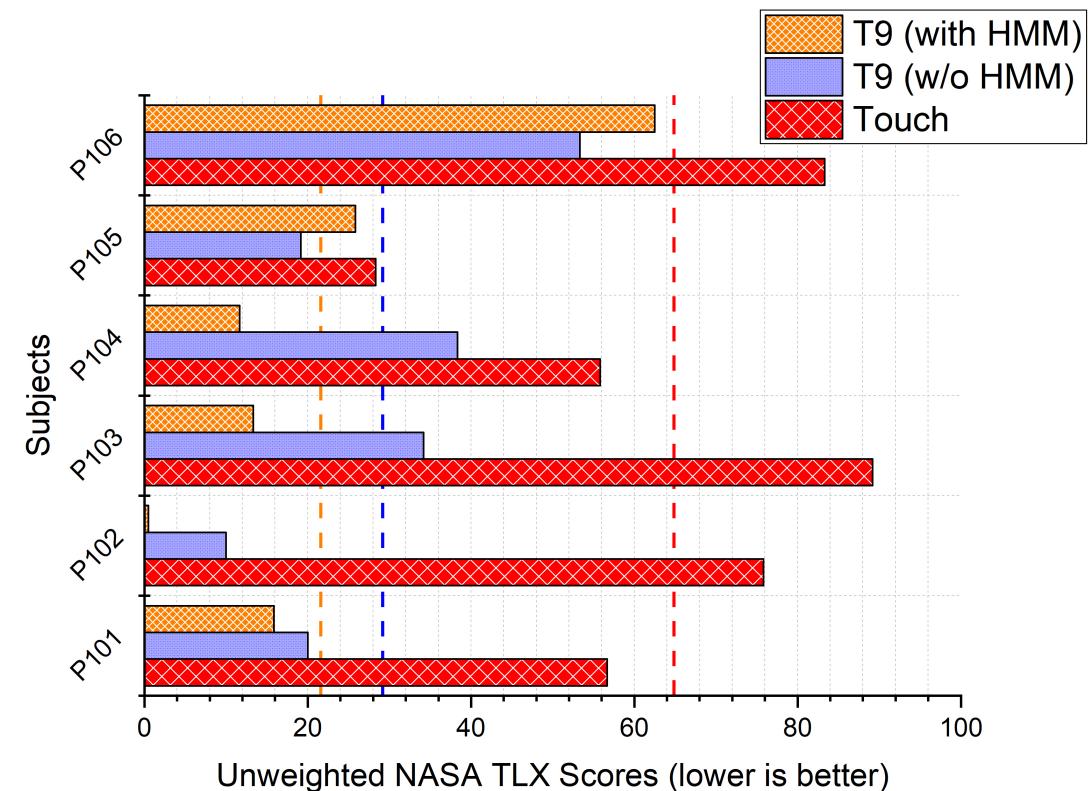
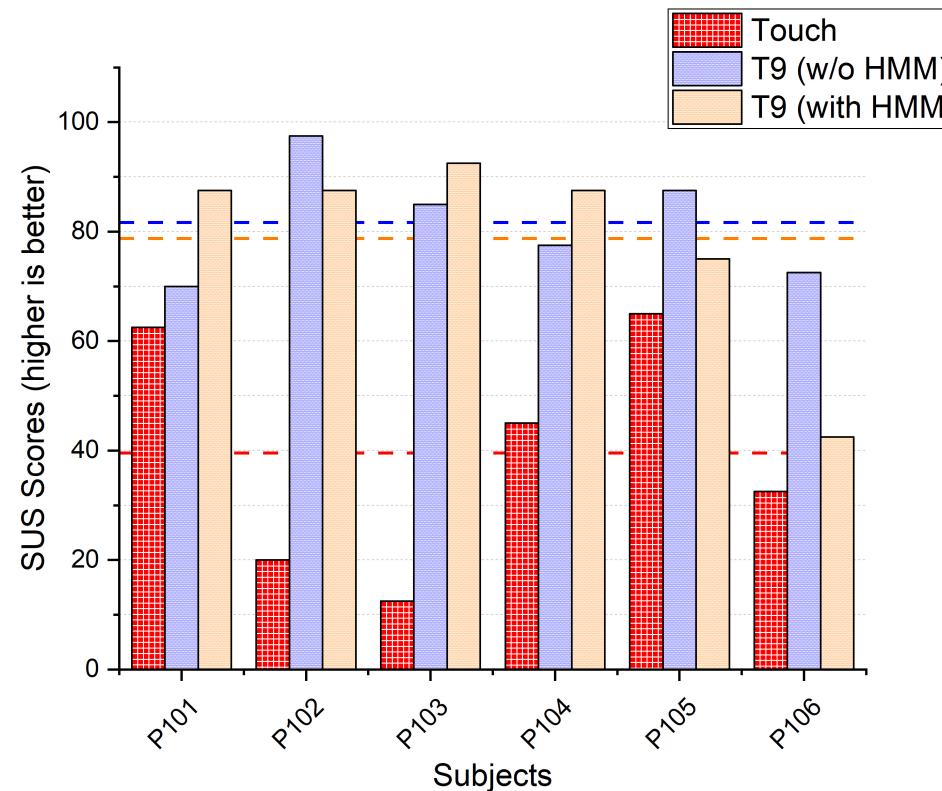
- Three scenarios:
  - Tiny touchscreen keyboard (1.8 X 2.3 cm)
  - HMM-T9 without AI
  - HMM-T9 with AI
- 6 participants – each type words from the given test set as per their choice for 2-3 minutes.
- Metrics:
  - Performance - Mean WPM (observed + literature);
  - Usability - System Usability Scale (SUS)
  - Workload - Unweighted NASA Task Load Index (TLX)
  - General and suggestive - questionnaire and rubric-based rating
- Statistical tests (Shapiro-Wilk, Friedman and Wilcoxon Signed Rank) to strengthen observed figures.

# User Study - WPM and Evidential Proof



- Shapiro-Wilk test: Gaussian distribution violation in “Touch” and “T9 (with HMM)”.
- Friedman test @  $p = 0.05$  reveals significant effect of method on WPM ( $\chi^2 = 10.33$ ,  $p > \chi^2 = 0.0057$ ) – **T9 system provides ~ 2x the speed of touchscreen baseline.**
- Friedman test @  $p = 0.05$  ( $\chi^2 = 2.67$ ,  $p > \chi^2 = 0.1025$ ) and post-hoc Wilcoxon signed ranks test reveals **negligible contribution of HMM on WPM**.
- **Observed WPM (8-9 WPM) in-line with larger T9 systems [1] - strong evidence to support observation.**
- **Miniaturized T9 keypad achieves similar WPM to phone T9 systems!**

# User Study - SUS and TLX Scores

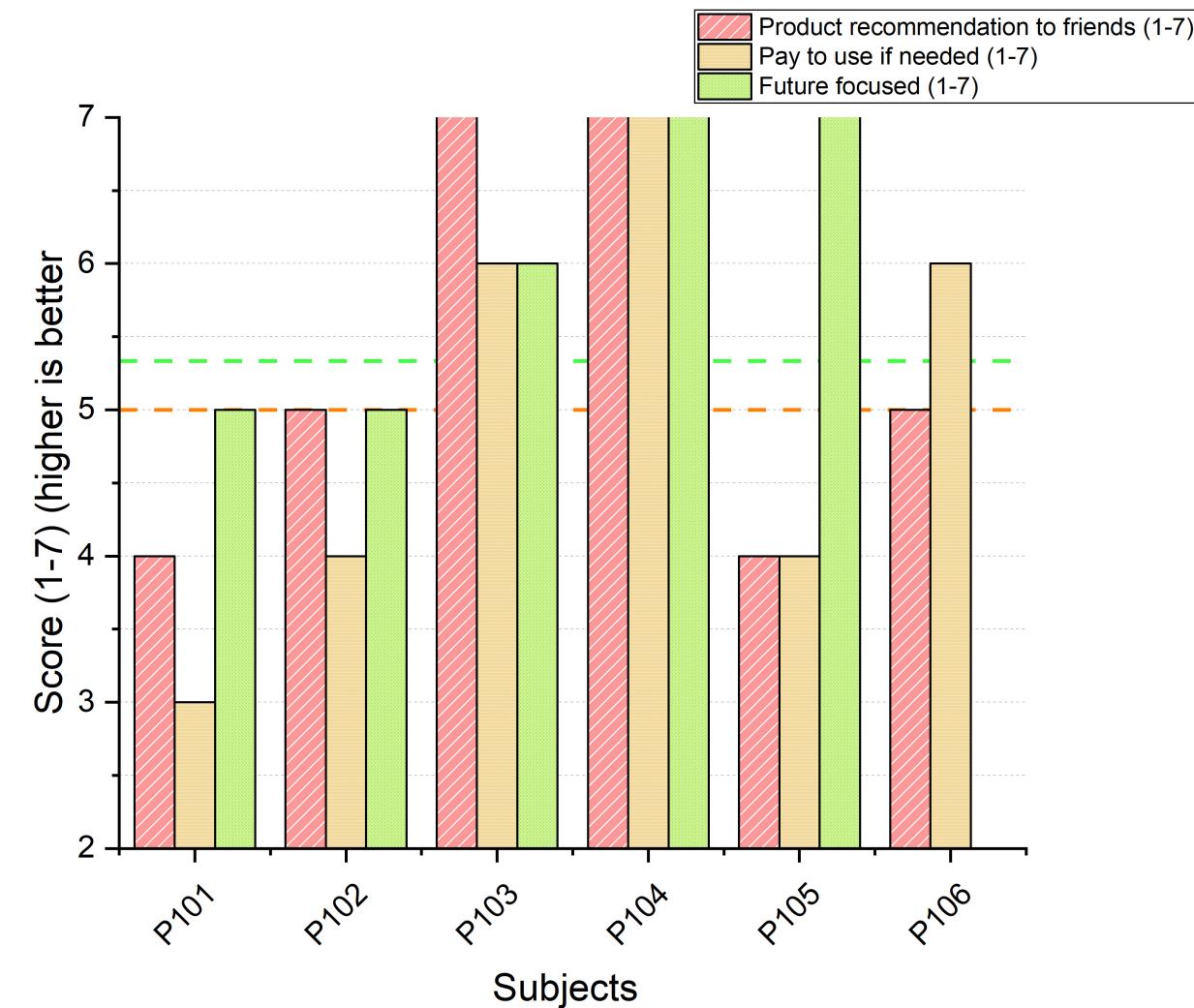


- Shapiro-Wilk test: Gaussian distribution violation only in “T9 (with HMM)” SUS score.
- Usability grade [1]** - Touchscreen: F, T9 w/o HMM: A, T9 with HMM: B.
- Workload category [2]** - Touchscreen: High, T9: Medium.
- Friedman test @  $p = 0.05$  reveals significant effect of method on SUS score ( $\chi^2 = 9$ ,  $p > \chi^2 = 0.011$ ) and TLX rating ( $\chi^2 = 9.33$ ,  $p > \chi^2 = 0.0094$ ) – **T9 system provides ~ 2x the usability and ~0.5x the workload compared to touchscreen baseline.**
- Friedman test @  $p = 0.05$  ( $\chi^2 = 0.2, 0.67$ ,  $p > \chi^2 = 0.65, 0.414$ ) and post-hoc Wilcoxon signed ranks test reveals **negligible contribution of HMM on SUS score and TLX rating.**

[1]. Sauro, Jeff. "Measuring usability with the system usability scale (SUS)." (2011).

[2]. Prabaswari, Atyanti Dyah, Chancard Basumerda, and Bagus Wahyu Utomo. "The Mental Workload Analysis of Staff in Study Program of Private Educational Organization." IOP Conference Series: Materials Science and Engineering. Vol. 528. No. 1. IOP Publishing, 2019.

# User Study - General Feedback



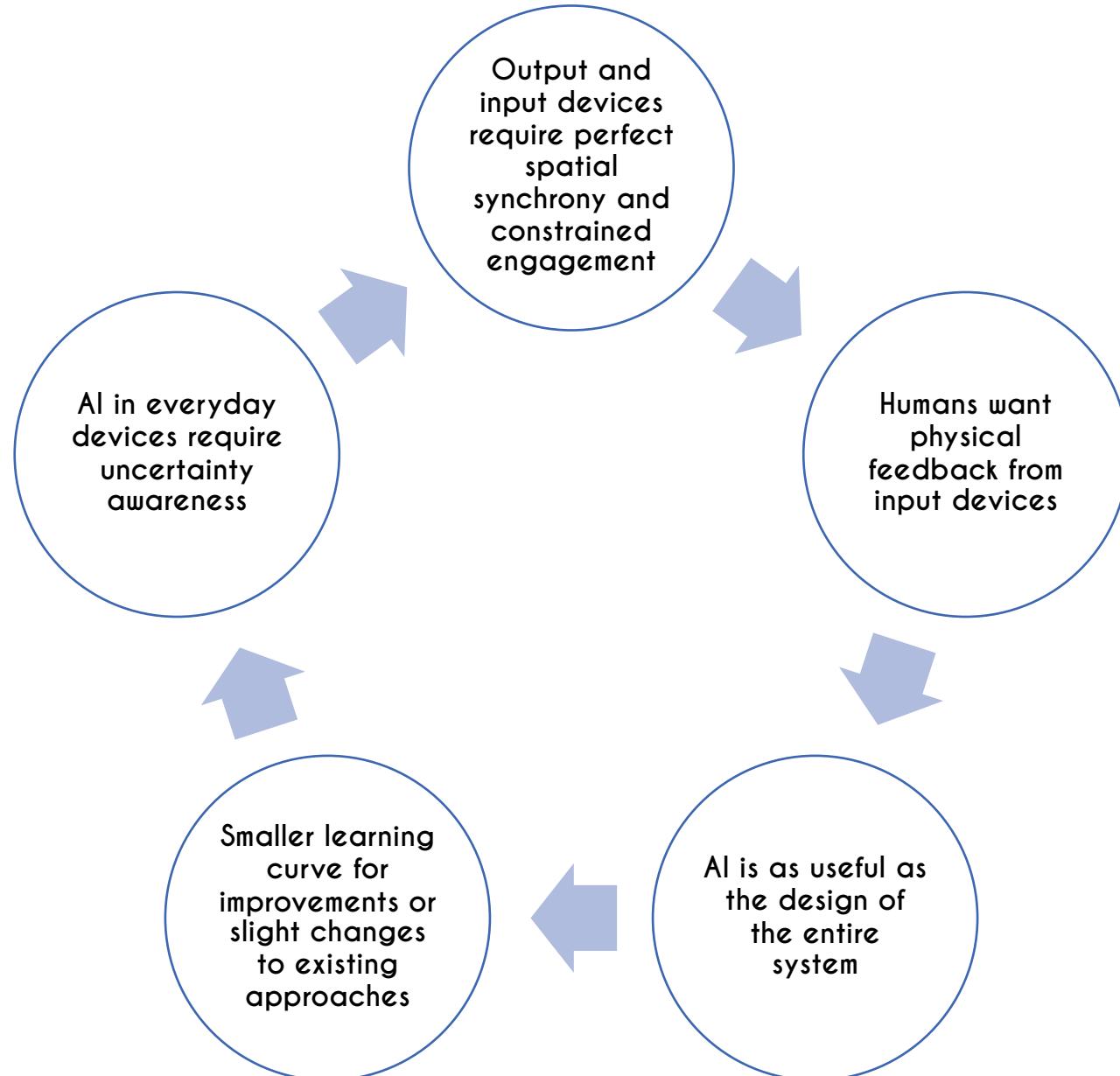
- Shapiro-Wilk test: No statistical bias found in user-rating.
- Recommendation, pay-to-use and future-usage rating:**  
~ 5 out of 7.

<i>Easy and Natural</i>	<p>“...keypad is clicky and the feedback makes it very easy to know that the key was pressed...”</p> <p>“...physical buttons much better than touch keys, clicking is helpful...easier to handle than touch...”</p> <p>“...natural layout, easy to learn...”</p>
<i>Useful and Fast</i>	<p>“...much faster than touchscreen keys...”</p> <p>“...prediction works well sometimes and very useful...”</p> <p>“...product can be learned quickly...”</p> <p>“...tactile feedback is useful...”</p>
<i>Novelty</i>	<p>“...integrating with HMM is novel...”</p> <p>“...the predictive word suggestions are novel...”</p> <p>“...capability to predict and complete words...”</p> <p>“...toggling suggestions with up and down key”</p>
<i>Usage-scenarios</i>	<p>“...wearable devices without touch screens, visually impaired people, low-cost fast typing portable keyboard...”</p> <p>“...battery constrained edge devices...”</p> <p>“...algorithm can be integrated with other devices, system can be used for smaller spaces...”</p>

# User Study – Suggestions

- AI needs further improvement:
  - HMM has no attention model to ignore confounding or erroneous inputs in its history.
  - Trained on small dataset with common words.
  - AI needs to learn context of the user input.
- Keypad and output device need to be closer to each other:
  - Hard to focus on the keypad and the screen simultaneously.
  - Keypad needs to be labelled well.
  - Too much to focus on for the AI-enabled setup (keypad, output, word suggestions, keypad labels)
- **Above points render AI not significantly helpful in our current demo.**
- Fix occasional bugs (e.g. poor handling of wrong inputs and backspace error).

# Lessons Learned



# Future Work

- Fix bugs in the current model, as well as the PCB.
- Explore sophisticated NLP models with uncertainty handling schemes (e.g. LSTM/RNN based models trained on larger datasets).
- Hardware and driver setup for a tiny OLED display near the keypad, capable of showing output and text predictions.
- On-device edge AI.

# THANK YOU

Code, results and hardware fabrication guide available for free at:

[https://github.com/swapnilsayansaha/ECE209AS\\_Fall2020](https://github.com/swapnilsayansaha/ECE209AS_Fall2020)