

Language-Conditioned Objectives for Task-Agnostic Preference Learning and Controller Updates

Stanford CS224N Custom Project

Kyeongwon Park

Department of Mechanical Engineering
Stanford University
parkkw@stanford.edu

Abstract

Preference-based personalization is attractive for gait assistance, but standard formulations typically assume a fixed human reward. We study whether language can help infer a user’s current objective and whether the resulting reward estimate is useful for controller updates. In Task A, models predict pairwise preferences over five gait outcomes from language and outcome differences under four feedback settings spanning single- vs. multi-attribute objectives and standard vs. harder language, where harder language introduces paraphrase, negation, and mild ambiguity. On the easy single-attribute / standard-language benchmark, TF-IDF and our structured model both solve the one-hot problem, indicating that shallow lexical features are sufficient when feedback is direct. In the multi-attribute / harder-language setting, however, the proposed model reaches 76.7% accuracy, outperforming the text-agnostic baseline by 5.0 points and TF-IDF by 13.3 points. A generic fine-tuned encoder+MLP remains weak throughout, suggesting that the gain comes from embedding language into an interpretable trade-off space rather than from text encoding alone. In Task B, the learned reward model improves mean true reward from 0.459 to 0.743 on in-split feedback and from 0.493 to 0.755 on held-out feedback. These results suggest that language can be a useful signal of context-dependent human objectives, while also highlighting the need for richer human reward models and real comparative language data.

1 Key Information

- **Mentor:** Caroline Choi
- **External collaborators:** None
- **Sharing project:** No
- **Late days used for this report:** 2

2 Introduction

Preference-based reward learning offers an appealing interface for human-in-the-loop adaptation: users need not hand-specify a scalar objective, and the system can instead infer what is desirable from lightweight feedback. This is particularly attractive in locomotion assistance and exoskeleton personalization, where repeated calibration is costly, individual responses differ substantially, and the notion of better assistance may vary across situations [Slade et al., 2022, Lee et al., 2023, Young and Ferris, 2016]. Yet much of the standard pipeline still treats the user’s reward as fixed. In practice, a person’s objective may shift with context: the same user may sometimes prioritize speed, sometimes lower effort, and sometimes reduced discomfort. If so, the scientific question is not only how to estimate preferences, but how to identify which aspect of the outcome matters at the present moment.

Language offers one possible signal for that hidden objective. Compared with a one-bit comparison, an utterance such as “I am getting out of breath” or “my steps feel uneven” can indicate why an outcome is preferred, not just which outcome wins. Recent work on comparative language feedback suggests that language can improve behavior optimization and reward learning [Yang et al., 2024]. Here we ask a narrower question: can language help recover a user’s current objective when that objective is context-dependent and may involve trade-offs among multiple outcomes? This question matters because if the optimizer is aligned with the wrong objective, additional preference data may still lead to systematically poor personalization.

Following the scientific framing of our proposal, we investigate three questions. **(Q1) Goal inference from text:** can a model map natural-language feedback t to a representation that improves prediction of $P(A \succ B \mid \phi_A, \phi_B, t)$ relative to text-agnostic preference learning? **(Q2) Robustness under objective shift:** does text conditioning remain useful under paraphrase, negation, mild ambiguity, and sparse multi-attribute trade-offs? **(Q3) Control usefulness:** when the learned reward model is used to score candidate controller updates, does it support better local search than a text-free alternative? These questions separate three issues that are often conflated: whether language contains objective information, whether that information survives more realistic linguistic variation, and whether the inferred reward is useful for downstream control.

To study them in a controlled way, we use a synthetic gait domain with five interpretable outcome dimensions and two linked tasks. **Task A** is language-conditioned preference modeling: given a feedback utterance and a pair of outcome summaries, predict which outcome the user prefers. **Task B** is language-conditioned controller update selection: use the learned reward estimate to score candidate control-law updates in simulation. Across these tasks, the experiments suggest a consistent picture. Language can help when the active objective is context-dependent, but the benefit is strongest when the model represents language as an explicit trade-off over interpretable outcomes rather than as an unstructured text feature. At the same time, the study is intentionally limited: the outcome space is designed rather than discovered, and the language is synthetic. We therefore present the results as a proof of concept for language-conditioned personalization, not as a complete account of human reward.

3 Related Work

Our work sits at the intersection of reward learning from human feedback, language-conditioned behavior improvement, and human-in-the-loop locomotion personalization.

Preference learning and reward inference. Bradley–Terry style paired-comparison models provide the classical probabilistic foundation for preference learning [Bradley and Terry, 1952]. In modern machine learning, preference feedback is often used to learn reward models or policies when explicit scalar rewards are unavailable [Christiano et al., 2017, Biyik and Sadigh, 2018]. These methods demonstrate that pairwise comparisons are practical and information-efficient, but they also make clear that one bit of supervision per query is limited when the candidate behaviors are close or when the relevant objective changes over time.

Language as a reward signal. The clearest inspiration for our project is the comparative-language framework of Yang et al. [2024], which uses language to improve trajectories and learn a reward predictor. Their work demonstrates that language can be more expressive than pure pairwise labels, but the feedback is still tied to a fixed underlying notion of improvement through an additive latent update. We instead study language as an explicit indicator of which objective is currently active. This framing is also related to context-aware preference modeling, which argues that preferences should be viewed as conditional rather than globally fixed [Pitis et al., 2024]. To our knowledge, however, prior language-conditioned reward-learning work has not directly studied sparse *multi-attribute* objective inference, where a single utterance may refer to multiple outcome dimensions and implicitly encode a trade-off among them. That multi-attribute extension is central to our setting.

Human-centered locomotion and personalization. Our synthetic task is motivated by real locomotion personalization problems. In exoskeleton control, it is generally not known in advance which policy will be best for a given person, because the human response depends on user-specific neuromuscular dynamics and interaction with the device [Zhang et al., 2017]. Human-in-the-loop op-

timization addresses this uncertainty by optimizing assistance directly on measured human outcomes, and has successfully discovered policies that improve metabolic cost in large controller spaces [Slade et al., 2022]. But metabolic cost is not the whole objective: broader locomotion and exoskeleton literature emphasizes additional factors such as comfort, stability, joint loading, symmetry, and task demands [Young and Ferris, 2016]. More recent work therefore asks users directly which assistance they prefer and performs preference-based personalization [Lee et al., 2023]. Much of this literature, however, still assumes that the relevant reward is fixed over the optimization window. Our project asks whether language can help resolve that objective mismatch by explicitly communicating the user’s current goal.

Text representations and baselines. We compare both shallow and neural text models. TF-IDF remains a strong lexical baseline for controlled language tasks [Salton and Buckley, 1988, Manning et al., 2008]. For encoder-based models, we use a pretrained Transformer encoder in the BERT family [Vaswani et al., 2017, Devlin et al., 2019]. Our main interest is not beating every possible language model, but rather asking whether an *interpretable structured head* provides a useful inductive bias for low-data goal inference.

4 Approach

4.1 Problem formulation

Each comparison consists of two candidates summarized by five normalized outcome features,

$$\phi_A, \phi_B \in [0, 1]^5, \quad (1)$$

and a feedback utterance t describing the user’s current complaint or desired change. The five outcomes are gait speed, metabolic cost, knee joint load, gait symmetry, and overall muscle activity. To make all dimensions point in the same semantic direction, we convert raw outcomes to a *benefit* representation in which larger is always better:

$$\text{benefit}(\phi) = [\phi_{\text{speed}}, 1 - \phi_{\text{met}}, 1 - \phi_{\text{knee}}, \phi_{\text{sym}}, 1 - \phi_{\text{muscle}}]. \quad (2)$$

We then define

$$\Delta\phi = \text{benefit}(\phi_A) - \text{benefit}(\phi_B), \quad (3)$$

and model the preference probability as

$$P(A \succ B \mid \phi_A, \phi_B, t) = \sigma(f_\theta(t, \Delta\phi)). \quad (4)$$

The central question in Task A is whether text helps identify a context-dependent reward over the outcome dimensions.

4.2 Proposed model: text-to-weights structured head

Our proposed model initializes its text encoder from pretrained DistilBERT [?], but the encoder is not used as a frozen feature extractor. In the neural models, the encoder parameters are fine-tuned jointly with the Task A preference objective, so the comparison is not between “pretrained text features” and “learned text features.” Rather, the key distinction is architectural: whether the language representation is read out through a head that respects human reward structure. Concretely, the encoder produces a text embedding $h(t)$, and a small feed-forward head maps that embedding to an explicit reward-weight vector over the five benefit dimensions:

$$\hat{w}(t) = \text{softmax}(g_\theta(h(t))) \in \Delta^5, \quad (5)$$

where Δ^5 denotes the probability simplex. The preference score is then

$$f_\theta(t, \Delta\phi) = s \hat{w}(t)^\top \Delta\phi + b, \quad (6)$$

with a learned global scale $s > 0$ and bias b .

This structure is deliberate. TF-IDF uses a fixed sparse lexical feature vector, while the encoder+MLP baseline uses the same pretrained backbone but leaves the mapping from language to preference completely generic. Our model instead forces the network to express language as a trade-off over interpretable gait outcomes. That inductive bias is appropriate for our scientific question: if human feedback reflects outcome-based trade-offs, then the language model should expose those trade-offs explicitly rather than bury them in an opaque classifier. The goal is therefore not only predictive accuracy, but also a representation whose failure modes are meaningful in terms of human reward.

4.3 Baselines

We evaluate three baselines. **Text-agnostic Bradley–Terry** is logistic regression on $\Delta\phi$ only and therefore assumes a single global reward vector. **TF-IDF interaction** builds features of the form $\text{TFIDF}(t) \odot \Delta\phi_j$ for each dimension j and fits logistic regression; this is a strong shallow baseline for any model that is approximately linear in bag-of-ngrams. **Encoder+MLP** concatenates a DistilBERT text embedding with $\Delta\phi$ and predicts the logit through a small MLP. All dataset generation, the structured text-to-weights head, training code, and Task B simulator were implemented by us. We reuse standard libraries for DistilBERT, PyTorch, and scikit-learn.

4.4 Task B controller-update loop

Task B reuses the learned reward model as a scoring function for controller updates. Let $\theta_c \in \mathbb{R}^5$ denote the current controller parameters and let $\hat{g} : \theta \mapsto \phi$ be a known surrogate mapping from controller parameters to the five human outcome features. Given feedback text t , the reward model outputs $\hat{w}(t)$, and the next controller is chosen from a feasible local candidate set $\mathcal{C}(\theta_c)$ by

$$\theta_{\text{next}} = \arg \max_{\theta' \in \mathcal{C}(\theta_c)} \hat{w}(t)^\top \text{benefit}(\hat{g}(\theta')). \quad (7)$$

In the final simulation run, each episode aggregates two feedback utterances by averaging the predicted weight vectors before renormalizing to the simplex. This design follows the same objective-conditioned update loop specified in the project plan: language determines the current objective, and controller updates are selected by scoring feasible local changes under that objective.

5 Experiments

5.1 Data

Task A datasets. We built a synthetic gait domain with five outcomes inspired by exoskeleton assistance and locomotion personalization [Young and Ferris, 2016, Slade et al., 2022]. The original dataset contains 300 comparisons (240 train / 60 test). We intentionally keep each setting small enough to mimic a practically limited preference-query regime while still allowing controlled comparisons across feedback conditions. For each example, we first sample a plausible base gait outcome vector with mild correlations among speed, effort, knee load, and symmetry. We then construct a controlled trade-off pair in benefit space: one *objective* feature favors one candidate, one *distractor* feature favors the other with similar magnitude, and the remaining dimensions contain small noise. This construction is important because it removes any single globally optimal reward vector; without the text, there is no consistent way to decide which candidate should win. Labels are deterministic oracle preferences generated from the active objective dimension.

Text comes from an English goal-text bank with 250 one-sentence utterances per objective. The bank was designed to be leak-free: no candidate identifiers, no numeric magnitudes, and no direct references to which option should win. We also enforced single-objective semantics in the original set so that each sentence describes exactly one desired outcome. This lets us cleanly test whether the model can recover the active reward dimension from natural language rather than from accidental label leakage.

We report results on four Task A feedback settings. **Single-attribute / standard-language** is the clean one-objective benchmark. **Single-attribute / harder-language** keeps the same numeric pairs, labels, and one-hot objectives as the original set but replaces the text with balanced paraphrase, negation, and mild ambiguity; this isolates language robustness while keeping the underlying preference geometry fixed. **Multi-attribute / standard-language** extends the objective from one-hot to sparse two-feature mixtures, with 100 balanced 50/50 samples and 200 dominant 70/30 samples spanning 30 objective classes. **Multi-attribute / harder-language** keeps the same multi-attribute numeric backbone and overlays harder language. All four settings contain 300 examples and use the same 240/60 split size.

Task B simulation. Task B assumes the Task A model has already been trained. We then place that model inside a simulated control loop with a known synthetic surrogate from 5D controller parameters to 5D outcome vectors. The final run uses the checkpoint trained on the *multi-attribute / harder-language* setting. For each episode, we sample a hidden true reward and two feedback

Table 1: Reward-recovery metrics for the proposed model.

Dataset	Cosine	Supp. Prec.	Supp. Rec.	Supp. F1	Argmax
Single / standard	1.000	1.000	1.000	1.000	1.000
Single / harder	1.000	1.000	1.000	1.000	1.000
Multi / standard	0.841	0.972	0.500	0.657	0.767
Multi / harder	0.833	0.958	0.508	0.658	0.817

utterances from the same split of that setting, estimate the reward from language, generate a local pool of candidate controllers near the current controller, and greedily select the highest-scoring feasible candidate. We evaluate 50 episodes with 8 update steps each and report results separately when the feedback texts come from the training split and the held-out test split.

5.2 Evaluation

For Task A we report **pairwise accuracy** and **negative log-likelihood (NLL)** on the held-out test set. Accuracy measures ranking correctness; NLL additionally reflects confidence and is the natural metric for a Bradley–Terry-style probabilistic model. We also compute **Brier score** and **10-bin expected calibration error (ECE)** to diagnose calibration. Because the synthetic datasets provide the true latent reward weights w_{true} , we can directly measure reward recovery for the proposed model using weight ℓ_1 , cosine similarity, support precision/recall/F1, argmax accuracy, and true-support mass.

For Task B we focus on whether the selected controllers actually improve the hidden true reward. The final run logs **mean start reward**, **mean final reward**, **mean gain**, **episode win rate** (fraction of episodes with positive gain), and **per-step win rate** (fraction of update steps that improve reward). Because the final stored run evaluates only the learned language-conditioned controller rather than a second Task B baseline, we use start-to-final improvement and held-out feedback generalization as the main evidence of control usefulness.

5.3 Experimental details

For TF-IDF we use word unigrams and bigrams with at most 50k features and logistic regression with $C = 10$. Encoder-based Task A models use `distilbert-base-uncased`, maximum sequence length 96, AdamW with learning rate 2×10^{-5} , batch size 32, validation fraction 0.15, and 5 training epochs. The random seed is 224. The proposed model uses the same encoder as the encoder+MLP baseline, but replaces the MLP scorer with the structured softmax-to-weights head described above.

The Task B final run uses the proposed checkpoint whose Task A test accuracy on the *multi-attribute / harder-language* setting is 0.767 with NLL 0.632 and weight cosine 0.833. Each Task B episode uses 2 feedback utterances, 8 update steps, a candidate pool of size 96, Gaussian candidate perturbation scale 0.25, and controller bounds of $[-1.5, 1.5]^5$. These values are intended to create a local-search regime: the model is not allowed to jump arbitrarily in controller space, so useful reward estimates must guide a sequence of myopic but improving updates.

5.4 Results

Figure 1 and Table 2 show a clear progression in difficulty across the four feedback settings. The single-attribute / standard-language benchmark is essentially a clean one-hot identification problem; in this regime, both TF-IDF and the proposed model succeed perfectly (1.000 accuracy), suggesting that when feedback is direct and the latent reward is atomic, shallow lexical cues are sufficient.

The single-attribute / harder-language setting isolates language robustness because the numeric comparisons are unchanged while the text is rewritten with paraphrase and negation. Here, the proposed model remains perfect (1.000), whereas TF-IDF degrades to 0.783, consistent with the structured head relying on more than fixed surface n-grams. The multi-attribute settings change the problem more fundamentally, requiring the model to infer a sparse trade-off. In the standard-language condition, the text-agnostic baseline rises above chance (0.717), indicating some globally useful structure in the outcome differences, but the proposed model remains the strongest predictor (0.850).

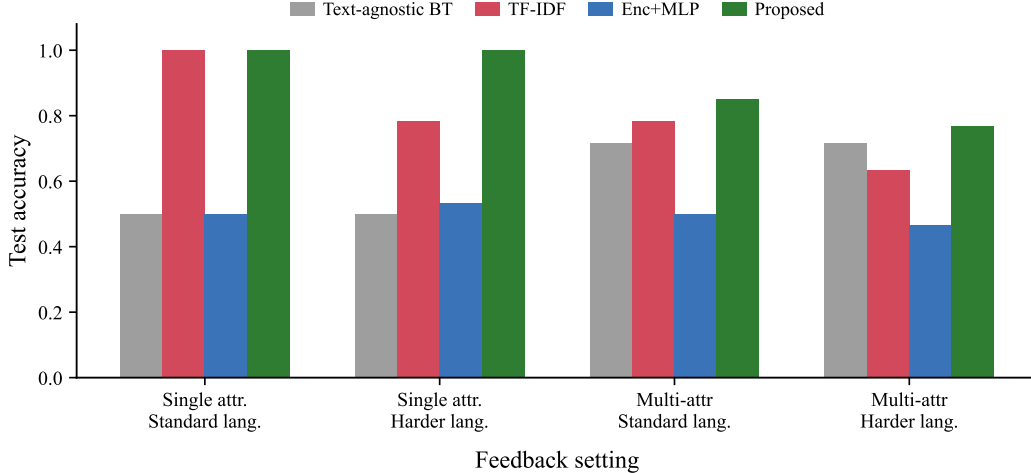


Figure 1: Task A test accuracy across the four feedback settings. The proposed structured model matches TF-IDF on the easiest benchmark, remains robust under harder language, and performs best on the multi-attribute settings.

Table 2: Task A test performance. Each cell reports accuracy / NLL.

Dataset	BT	TF-IDF	Enc+MLP	Proposed
Single / standard	0.500 / 0.697	1.000 / 0.360	0.500 / 0.697	1.000 / 0.574
Single / harder	0.500 / 0.697	0.783 / 0.591	0.533 / 0.690	1.000 / 0.574
Multi / standard	0.717 / 0.616	0.783 / 0.592	0.500 / 0.692	0.850 / 0.604
Multi / harder	0.717 / 0.616	0.633 / 0.651	0.467 / 0.694	0.767 / 0.632

The hardest case is multi-attribute / harder-language, where latent trade-off inference and indirect language appear together. All text-conditioned models degrade here—the proposed model to 0.767, TF-IDF to 0.633, and the encoder+MLP to 0.467—but the structured model degrades least and remains the superior approach.

Table 1 clarifies the main failure mode. On the two single-attribute settings, the predicted reward vector is essentially exact: cosine similarity, support precision/recall/F1, and argmax accuracy are all 1.000. On the multi-attribute settings, however, recovery becomes only partial. For the *multi-attribute / standard-language* setting, the proposed model reaches weight cosine 0.841, support precision 0.972, support recall 0.500, support F1 0.657, and argmax accuracy 0.767. For the *multi-attribute / harder-language* setting, the corresponding numbers are 0.833, 0.958, 0.508, 0.658, and 0.817. These values show that the model usually places mass on the correct support, but often fails to recover the full mixture cleanly. Task B nevertheless shows that these imperfect reward estimates can still be operationally useful.

Figure 2 and Table 3 reports the final Task B simulation results. Even though the controller-selection model is trained on the hardest Task A setting, it consistently improves reward in simulation. On training-split feedback, the mean true reward increases from 0.459 to 0.743, with positive final gain in 98% of episodes. On held-out feedback, the mean reward increases from 0.493 to 0.755, with 92% episode win rate. The gap between train and held-out feedback is modest, suggesting that the learned reward model transfers reasonably well to unseen utterances drawn from the same multi-attribute / harder-language feedback distribution. Note that the other baselines are pairwise preference classifiers and therefore do not directly support this downstream reward-guided optimization setting. Learning latent reward values (proposed model) is useful as it enables the direct simulation of policy effects.

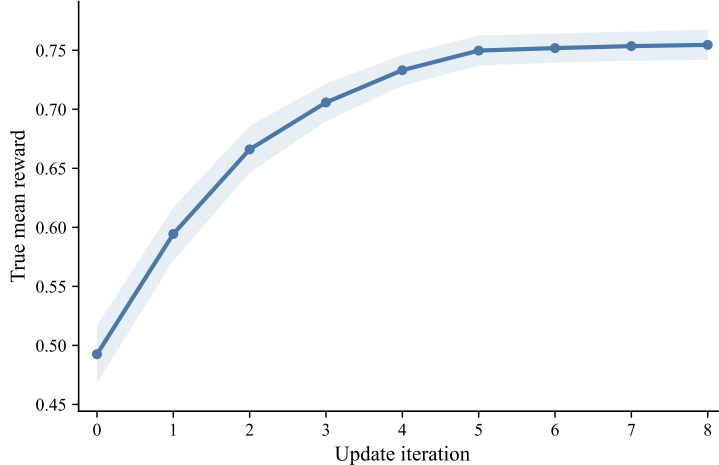


Figure 2: Task B mean true reward over controller-update iterations under held-out feedback and the proposed model. The language-conditioned reward estimate supports steady improvement over the local search trajectory. Note that the other baselines are pairwise preference classifiers, not absolute reward models, and therefore do not directly support this downstream reward-guided optimization.

Table 3: Task B controller-update results for the learned language-conditioned reward model.

Feedback split	Start reward	Final reward	Mean gain	Win rate	Per-step win
Train feedback	0.459	0.743	0.283	0.980	0.715
Held-out feedback	0.493	0.755	0.262	0.920	0.670

6 Analysis & Discussion

Language as a partial observation of human objective. The results suggest that language can help identify not only *which* option is preferred, but also *why* it is preferred. In this controlled setting, the benefit is most visible once the objective is no longer fixed to a single outcome dimension. That should not be interpreted as language fully specifying reward. Rather, language appears to provide a useful partial observation of the user’s current trade-off over interpretable outcomes.

Why structure seems to matter. The comparison between the encoder+MLP baseline and the proposed model is informative because both use the same fine-tuned DistilBERT encoder. The difference lies in the output space: the proposed model is forced to express language as a trade-off over outcome dimensions. This suggests that the improvement is not simply due to stronger text features, but to aligning the hypothesis class with the way preferences are generated in this domain. More broadly, language may be most useful when it is grounded in a representation of the outcome trade-offs that are actually available to the user.

What Task B adds. Task B provides a proof of concept for surrogate human-in-the-loop optimization. The learned reward estimate is not exact, yet it is good enough to rank nearby controller candidates and support monotonic improvement in a local search loop. This is encouraging because a practical system would rarely need a perfect global reward model; it would need a reward estimate that is sufficiently aligned with the user’s current objective to make better next-step decisions.

What should be interpreted cautiously. These experiments do not show that language alone solves personalization. The study uses a synthetic outcome space with five designed features, deterministic labels, and an oracle surrogate in Task B. The text-agnostic baseline also performs above chance on the multi-attribute settings, which indicates that the benchmark retains some global structure even without language. Task B should therefore be read as evidence that language-conditioned reward estimates can be useful, not as a demonstration that language is the only or dominant source of objective information.

What needs to be established next. Several assumptions should be tested before drawing stronger conclusions. The relevant human objective may involve latent factors outside the five outcomes used here. People may verbalize only a subset of what they care about, especially when objectives become richer or more context-dependent. And the language used in practice may differ from our LLM-augmented bank in tone, specificity, and consistency. A stronger scientific account will therefore require real comparative language from human users, broader or learned outcome spaces, calibrated uncertainty, and controller-selection baselines inside the Task B loop.

7 Conclusion

This project examined whether language can help infer a person’s *current* objective, rather than merely annotate a fixed reward. In our controlled gait setting, a structured language-to-trade-off model matched shallow lexical methods on the easiest one-hot benchmark, remained more robust under harder language, and performed best once objectives became multi-attribute. When inserted into a simulated local search loop, the resulting reward estimate was also sufficient to improve controller quality over successive updates.

The main limitation is that this remains a proof of concept: the reward space is hand-designed, the feedback is synthetic, and the controller surrogate is known. The next step is therefore not simply scaling the model, but testing the assumptions underneath it—which outcome dimensions matter to people, how those objectives are actually verbalized, and how uncertainty should be handled when reward estimates are used to drive control decisions. Answering those questions would move language-conditioned personalization from a promising mechanism toward a more scientifically grounded human-in-the-loop framework.

Team contributions

This was a single-person project. Kyeongwon Park designed the problem, implemented the data generators, models, and simulator, ran all experiments, analyzed the results, and wrote the report.

References

- Patrick Slade, Mykel J. Kochenderfer, Scott L. Delp, and Steven H. Collins. Personalizing exoskeleton assistance while walking in the real world. *Nature*, 610(7931):277–282, 2022.
- Ung Hee Lee, Varun S Shetty, Patrick W Franks, Jie Tan, Georgios Evangelopoulos, Sehoon Ha, and Elliott J Rouse. User preference optimization for control of ankle exoskeletons using sample efficient active learning. *Science Robotics*, 8(83):eadg3705, 2023.
- Aaron J Young and Daniel P Ferris. State of the art and future directions for lower limb robotic exoskeletons. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(2): 171–182, 2016.
- Zhaojing Yang, Miru Jun, Jeremy Tien, Stuart J Russell, Anca Dragan, and Erdem Bıyık. Trajectory improvement and reward learning from comparative language feedback. *arXiv preprint arXiv:2410.06401*, 2024.
- Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- Erdem Biyik and Dorsa Sadigh. Batch active preference-based learning of reward functions. In *Conference on robot learning*, pages 519–528. PMLR, 2018.
- Silviu Pitis, Ziang Xiao, Nicolas Le Roux, and Alessandro Sordoni. Improving context-aware preference modeling for language models. *Advances in Neural Information Processing Systems*, 37:70793–70827, 2024.

- Juanjuan Zhang, Pieter Fiers, Kirby A Witte, Rachel W Jackson, Katherine L Poggensee, Christopher G Atkeson, and Steven H Collins. Human-in-the-loop optimization of exoskeleton assistance during walking. *Science*, 356(6344):1280–1284, 2017.
- Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.