



What Makes a Model Breathe? Understanding Reinforcement Learning Reward Function Design in Biomechanical User Simulation

Hannah Selder

Center for Scalable Data Analytics and Artificial
Intelligence (ScaDS.AI)
Leipzig University
Leipzig, Germany
hannah.selder@uni-leipzig.de

Per Ola Kristensson

University of Cambridge
Cambridge, United Kingdom
pok21@cam.ac.uk

Florian Fischer

University of Cambridge
Cambridge, United Kingdom
fjf33@cam.ac.uk

Arthur Fleig

Center for Scalable Data Analytics and Artificial
Intelligence (ScaDS.AI)
Leipzig University
Leipzig, Germany
arthur.fleig@uni-leipzig.de

Abstract

Biomechanical models allow for diverse simulations of user movements in interaction. Their performance depends critically on the careful design of reward functions, yet the interplay between reward components and emergent behaviours remains poorly understood. We investigate what makes a model “breathe” by systematically analysing the impact of rewarding effort minimisation, task completion, and target proximity on movement trajectories. Using a choice reaction task as a test-bed, we find that a combination of completion bonus and proximity incentives is essential for task success. Effort terms are optional, but can help avoid irregularities if scaled appropriately. Our work offers practical insights for HCI designers to create realistic simulations without needing deep reinforcement learning expertise, advancing the use of simulations as a powerful tool for interaction design and evaluation in HCI.

CCS Concepts

• **Human-centered computing** → **User models.**

Keywords

Reward design, biomechanical models, deep reinforcement learning

ACM Reference Format:

Hannah Selder, Florian Fischer, Per Ola Kristensson, and Arthur Fleig. 2025. What Makes a Model Breathe? Understanding Reinforcement Learning Reward Function Design in Biomechanical User Simulation. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*, April 26–May 01, 2025, Yokohama, Japan. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3706599.3719699>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI EA '25, Yokohama, Japan

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1395-8/25/04

<https://doi.org/10.1145/3706599.3719699>

1 Introduction

Biomechanical simulations offer great potential for modelling and understanding human body movements. They can be used to predict muscle fatigue [8], anticipate potential injury risks [44], and guide the design of rehabilitation programs [37]. In the context of human-computer interaction (HCI), biomechanical models have been successful in predicting how users type on a virtual keyboard [20], point at and track moving mid-air targets [14, 23, 25], or play exergames in virtual reality (VR) [16]. Recently, visual perception models have been integrated with musculoskeletal dynamics [23], strengthening the value of biomechanical simulations as a basis for comprehensive user models.

However, the quality of biomechanical user simulations essentially relies on their ability to reliably capture task-relevant behaviours and motion patterns. In reinforcement learning (RL), the most promising and commonly used technique to forward-simulate musculoskeletal models [6, 23, 40], a key ingredient is an appropriate *reward function*. RL (and other optimal control methods such as Model Predictive Control [25] or the Linear Quadratic Gaussian Control [15]) is based on the assumption of rational behaviour: an *agent*, that is, a real or simulated entity capable of making decisions within a defined context, is assumed to observe their environment and ensure their actions align with their overarching goals. The reward function summarises the agent’s goals, and therefore may change as the task or context changes. For example, if the agent is to grasp a cup, or type a certain word on a keyboard, this should be reflected in the reward function.

While recent works have demonstrated the potential of RL-trained agents to simulate plausible human movement in various HCI contexts, e.g., [14, 23], they rarely communicate the amount of work spent tuning the reward function. Depending on the task’s complexity and specificity, identifying an appropriate reward function can be very challenging. Main challenges include the lack of general guidelines for novices, missing insights into the comparative performance of relevant reward components, and the difficulty of “trial-and-error” approaches due to extensive RL training times

when using state-of-the-art biomechanical models, which are typically in the order of 12–72 hours on modern workstations for testing a single reward function. This considerably limits the applicability of current biomechanical simulations to HCI and prevents further engagement with RL-based simulation frameworks.

This late-breaking work constitutes a first key step toward addressing current issues related to reward function design in biomechanical user simulations. We specifically consider the choice reaction task from [23], which is well-established in HCI. In this task, users are given several buttons and a monitor in front of them and need to click the button of the colour currently displayed on the monitor as fast as possible. This task requires non-trivial muscle coordination and skills essential to visuomotor interaction: colour vision, object recognition and aimed arm movements. For this choice reaction task, we provide a comprehensive and thorough evaluation of relevant reward terms through 60 trained policies. In particular, we address the following **research questions**:

RQ.1 (Plausibility) What combinations and relative weightings of reward function components (e.g., proximity, effort, and task completion bonuses) produce plausible human movement trajectories in a choice reaction task?

RQ.2 (Sensitivity) What are the sensitivities of interaction outcomes to variations in individual reward function components in a choice reaction task?

With **RQ.1** we aim to help researchers and practitioners who seek optimized reward functions. With **RQ.2** we aim to gain a better understanding of the difficulties inherent to reward function design and to derive guidelines for the development of composite reward functions.

Specifically, we **contribute to computational modelling of movement-based interaction** by providing

- a first systematic exploration of standard reward function components for a movement-based HCI task, and
- guidelines and first principles for reward function design in RL-based biomechanical user simulations.

2 Related Work

Biomechanical simulations are a beneficial tool for developing and validating HCI technologies [2, 16, 31]. They utilize dynamic models of the human body to predict movement during interaction. While early models were limited to calculating mechanical loads in static postures [1, 17, 46], advancements have led to physiologically increasingly accurate musculoskeletal models [12, 13, 28]. These models are integrated into user simulations to generate realistic movements [30, 36]. For example, Hwang et al. apply simulations with electrical muscle stimulation to create kinesthetic force feedback for virtual reality [22]. Biomechanical models have also been used to simulate human movements in interaction tasks [15, 16, 23, 25] and to analyse the cognitive aspects of interaction tasks [9, 10].

Deep RL has emerged as the go-to method for simulating movement-based interaction. For example, Fischer et al. use it to learn controlling the muscles of a state-of-the-art shoulder model for a mid-air pointing task. Ikkala et al. present *User-in-the-Box*, an RL-based simulation framework to generate task-specific movement trajectories based on the user’s visual and proprioceptive perception of the interaction environment. Other optimal feedback control

methods have been investigated for simulating movement-based interaction [15, 25, 29]. However, these have shown applicable for relatively low-dimensional control problems only, imposing severe restrictions on the complexity of the biomechanical models and tasks considered.

In RL-based biomechanical simulations, the design of the reward function is identified as a key factor in the effectiveness of the learning process [26]. Consequently, formulations of effective reward functions for specific simulated tasks have been examined [19, 23, 26, 34]. However, most of the proposed reward functions have been handcrafted for a specific task [5, 6, 23], limiting their generalizability across tasks and contexts. In addition, reward functions usually involve a trade-off between two or more opposing objectives, e.g., between accuracy and stability [27] or speed and accuracy [32]. In particular, composite reward functions typically include at least one “effort” term that penalizes large controls, which restricts the use of rapid and abrupt arm movements and ensures that available resources are used efficiently. Several effort cost models have been proposed and investigated from a motor control perspective [4, 8, 43, 45]. These models address the redundancy of movement problem, which refers to the fact that humans can perform tasks with an infinite number of different admissible joint trajectories [4]. This is because penalizing different behaviours, such as rapid and jerky arm movements, leads to different movement patterns [4, 18]. Furthermore, the role of an effort term in motor adaptation is explored in the empirical studies in [35, 48].

While different effort cost models have been proposed and investigated from a motor control perspective [4, 8, 43, 45], there exist no guidelines on how to design and balance these reward components in practical HCI tasks, especially in combination with complex musculoskeletal systems.

We therefore anticipate a strong need to explore the design of reward functions for realistic use cases of biomechanical models. In this work, we make a decisive step towards this goal by starting with a simple choice reaction task and analysing the individual and combined effects of different reward function components on RL-based learning of interactive body movement.

3 Methodology

In this work, we analyse the effect of different reward components on the predicted user strategies in an RL-based biomechanical simulation approach. We focus on the choice reaction task implemented in the *User-in-the-Box* (UitB) framework¹. The agent is provided with four different coloured buttons and a stimulus (one of the four colours) shown on a display in front of them. The task is to press the button of the displayed colour as fast as possible, within a maximum period of four seconds per trial. As soon as the correct button is pressed with a suitable force, the displayed colour switches and the next trial starts.

For each considered reward function, we train an RL policy within the *UitB* choice reaction environment following the procedure described in Ikkala et al. [23]. In particular, we use the default *MoBL Arms Model* [38] with 5 DoFs (three independent shoulder joints, elbow, wrist) and 26 muscles enabled, and provide visual, proprioceptive, and tactile information as input to the agent. Each

¹<https://github.com/User-in-the-Box/user-in-the-box>

episode starts with the arm hanging down (see Figure 2 (left)). Each policy trained for a given reward function can be used to simulate and predict user behaviour; we therefore denote a trained agent as *simulated user* in the following.

When designing the reward function, we focus on three components:

- The *completion bonus* component rewards task completion, e.g., similar to scores in games. While it can be simply a constant for tapping the right button, we integrate the many possibilities in the function $f_{\text{bonus}}(\cdot)$, where (\cdot) is a placeholder for all relevant function arguments.
- The *distance* component rewards the agent more the closer they get to the target. In the choice-reaction task, this means moving towards the right-coloured button. To incorporate the many mathematical formulations, we introduce the function $f_{\text{distance}}(\cdot)$.
- The *effort* component is very versatile: Designers can choose, e.g., to penalize jerky movements, or reward movements that require lower energy. We encompass the possibilities in the function $f_{\text{effort}}(\cdot)$.

To evaluate the intricacies of how the individual components work independently and in conjunction, for each component we introduce respective weights $w_{\text{distance}}, w_{\text{effort}}, w_{\text{bonus}} \geq 0$. In total, the most generic reward function amounts to

$$r_t = w_{\text{bonus}} \cdot f_{\text{bonus}}(\cdot) - w_{\text{distance}} \cdot f_{\text{distance}}(\cdot) - w_{\text{effort}} \cdot f_{\text{effort}}(\cdot). \quad (1)$$

If we set $w_{\text{effort}} = 0$ or $w_{\text{distance}} = 0$, we speak of *zero effort* or *zero distance*, respectively.

3.1 Reward Components

Each simulated user was trained for 35M steps, as we observed that further training beyond this point did not yield additional learning or improvements (the UitB framework suggests a default of 50M training steps).

For the *completion bonus*, we follow [23] and consider different constant values $b \geq 0$, i.e.,

$$f_{\text{bonus}}(\cdot) = \begin{cases} b, & \text{if the correct button is pressed,} \\ 0 & \text{else.} \end{cases} \quad (\text{Bonus})$$

We investigate three different *distance* reward functions, each based on the distance between the index finger and the surface of the currently desired button, *dist*, as measured by a MuJoCo distance sensor [41]:

- (1) The (absolute) value of the MuJoCo distance sensor:

$$f_{\text{distance}}(\text{dist}) = |\text{dist}| \quad (D_{\text{absolute}})$$

- (2) The squared distance, which has been successfully used in RL tasks [24, 33]:

$$f_{\text{distance}}(\text{dist}) = \text{dist}^2 \quad (D_{\text{squared}})$$

- (3) An exponential transformation of the distance, as used in [23]:

$$f_{\text{distance}}(\text{dist}) = \frac{1 - e^{-10 \cdot \text{dist}}}{10} \quad (D_{\text{exponential}})$$

We also compare different *effort* models. The first one, denoted as *EJK* in the following, was first presented in [7] to simulate

realistic arm movements and consists of three components. This is motivated by the observation that combining multiple effort terms can improve the plausibility of generated movements [4, 47]. Its components penalize the mean value of the muscle stimulation commands (r_{energy}), the jerk, i.e., the change in joint acceleration (r_{jerk}), and the total work done by the shoulder and elbow (r_{work}) in terms of angular velocities and torques. These components are normalized and weighted by coefficients c_1, c_2 and c_3 , respectively, resulting in the following effort model:

$$f_{\text{effort}}(r_{\text{energy}}, r_{\text{jerk}}, r_{\text{work}}) = \frac{c_1 r_{\text{energy}} + c_2 r_{\text{jerk}} + c_3 r_{\text{work}}}{c_1 + c_2 + c_3} \quad (\text{EJK})$$

Furthermore, we consider the three effort models from [25] (*DC*, *CTC*, and *JAC*), where their suitability to predict mid-air pointing movements using a non-RL optimization method (MPC) was examined. All three models include a penalty for large muscle stimulation commands u , motivated by the fact that humans seek to minimize their control effort during movement [42]. In the following models, this muscle effort term is penalized in the norm, whereas the EJK model considers its mean in r_{energy} . The *DC* effort model only consists of this penalty term, weighted by a coefficient c_1 :

$$f_{\text{effort}}(u) = c_1 \|u\|^2 \quad (\text{DC})$$

The *CTC* model adds a penalty on large changes in commanded torque τ , which is the torque at the joints that directly results from the controlled muscle activations. This term is motivated by a study from Wada et al., where the minimum commanded torque *change*, i.e., the derivative of τ , criterion was found to be the most effective in explaining the temporal characteristics of actual hand trajectories. The *CTC* model is described as follows:

$$f_{\text{effort}}(u, \dot{\tau}) = c_1 \|u\|^2 + c_2 \|\dot{\tau}\|^2 \quad (\text{CTC})$$

Similarly, the *JAC* model adds a penalty on large joint accelerations x_{qacc} , thus avoiding "jerky" movements. This effort term was introduced in [43] and later found to provide the most comprehensive explanation of mid-air pointing movements [25]. In contrast to the EJK effort model, this model penalizes the acceleration values themselves instead of their changes. The resulting *JAC* model is defined as follows:

$$f_{\text{effort}}(u, x_{\text{qacc}}) = c_1 \|u\|^2 + c_2 \|x_{\text{qacc}}\|^2 \quad (\text{JAC})$$

4 Results

This section presents the results of training models with various reward functions. A qualitative evaluation of model behaviours, based on evaluation videos, complements the quantitative success rates and completion times shown in Figure 3. The parameter values for all considered conditions (denoted as IDs in the following) are detailed in Appendix A. For a visual representation of these results, we refer to the video figure attached in the supplementary material.

4.1 Qualitative Results

Models trained with task completion bonus only did not learn to press all four buttons equally (ID: 31). Instead, only the green button is successfully reached. For the remaining three colours, noisy and non-directed arm movements are generated, as shown in Figure 1 (left). Increasing the bonus value did not lead to fundamentally different movement trajectories (ID: 32).

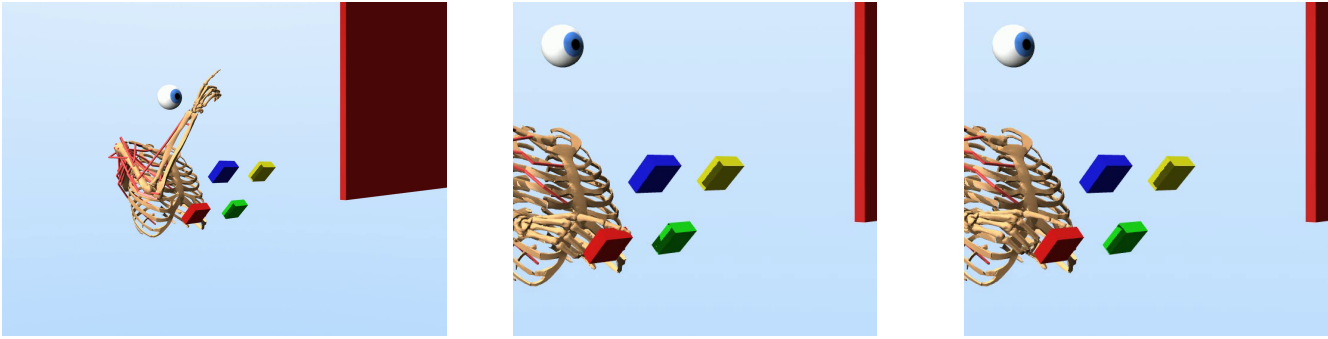


Figure 1: Comparison of movement patterns: only including the completion bonus leads to arbitrary movements (left); a combination of distance and effort rewards may incentivize hitting the button from the side or stopping immediately below the target (middle); a combination of completion bonus and distance rewards leads to reasonable arm movements and successful button clicks (right).

As a next step, we added a distance reward term. As expected, the distance reward helps "guiding" the RL policy towards states in which the fingertip is close to the desired button, i.e., the agent learns to identify the correct button and moves towards it. The choice of the distance reward function has an impact on the learned strategy. For the squared and exponential distance function, the simulated user tries to press the red button with the proximal phalanx of the index finger, i.e., the lower part of the finger close to the back of the hand, which often requires multiple attempts (IDs: 25, 27). This behaviour was not observed for the absolute distance (ID: 26); here, the red button is regularly (and most of the time successfully) pressed with the fingertip (see Figure 1 (right)). The remaining three buttons are regularly approached with the fingertip independent of the chosen distance function.

Using distance and effort rewards only, i.e., omitting the completion bonus in the reward function, does not lead to successful movements (IDs: 4-6, 10-12, 16-18, 22-24). Specifically, the model hits the buttons sideways instead of pressing them correctly, consequently failing to fulfil the task, see Figure 1 (middle).

Finally, we were interested in the effects of including an effort model. For the CTC effort model, we again observe a difference between the absolute distance model and squared or exponential distance; for the former, the model successfully presses all buttons, while for the latter two, the trained model is unable to press the upper two of the buttons and remains at the lower two buttons instead (IDs: 13-15). For the DC effort model with squared or exponential distance rewards, similar behaviour as without effort term was observed, i.e., the red button specifically is pushed with the back of the hand (IDs: 19, 21). For the remaining effort models, i.e. JAC and EJK, all three distance terms result in reasonable behaviour for each button (IDs: 1-3, 7-9). When the completion bonus is omitted (i.e., distance and effort rewards only (IDs: 4-6, 10-12, 16-18, 22-24)), the effort models JAC, CTC and DC demonstrated superior performance with the absolute value of the distance added directly, although they are only able to hit the buttons on the side. With the exponential or squared form of the distance rewards, the models are unable to hit the buttons, even from the side, and struggle with hitting the yellow button, which is the furthest away.

Without a completion bonus and using exponential distance rewards (IDs: 4, 22), the DC and EJK models result in the simulated user pressing nearby buttons from the side (the same applies to the JAC and CTC effort models with linear distance). The JAC model with exponential distance rewards exhibits a strategy of hitting buttons from below unless the next button is directly underneath (ID: 10). With increased effort weights, distinctions become more apparent, as can be inferred from Figure 2 (IDs: 45, 53, 49). The JAC model causes the arm to remain extended, whereas the DC model bends the arm and raises it towards the buttons. The CTC effort generates minimal rotational movements. However, when a completion bonus is added, most visual differences diminish, except that the DC model causes the hand to rotate when pressing the green button.

A reward function combining only effort models (e.g., EJK) and the completion bonus fails to initiate movement, further emphasizing the need for a complementary, task-specific "guidance" term in the reward function (IDs: 33-35).

In addition, we found that the choice of the effort weight is critical for task performance. For example, with the EJK effort model, exponential distance term and completion bonus, large effort weights prevent the model from pressing all buttons, limiting it to those closest to the initial position (ID: 36). Reducing the effort weight enables the model to press more buttons, eventually achieving full task completion. However, further reductions lead to inconsistent performance, with increased failed attempts on the red button (ID: 43). All these effects were observed independently of the selected effort model. In addition, the JAC effort model is also sensitive towards the relative scaling of the two effort components (i.e., the choice of c_1 and c_2 in JAC). Increasing the weight for the joint acceleration costs c_2 results in movements where the hand is placed close to the centre of the four buttons and only one of the four buttons is hit successfully (whereas, with default weight, the simulated user is able to hit all buttons (IDs: 46, 47)).

4.2 Quantitative Results

Figure 3 shows the success rates and average task completion times of 35 trained policies (see Table 1 in the appendix), calculated from

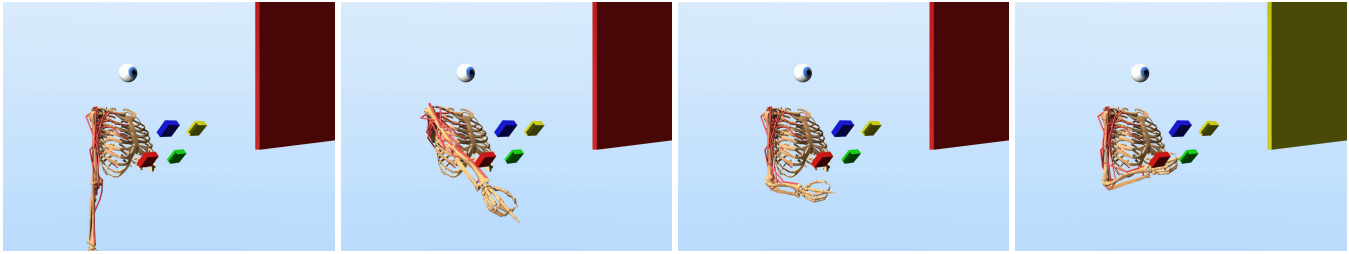


Figure 2: Comparison of movement patterns of different effort models, from left to right: CTC model with no movement, JAC model with extended arm, DC model with bent arm, and EJK model remaining on the lower buttons.

5 episodes with 10 required button clicks each. It is evident that models trained without the completion bonus (pluses) consistently fail to achieve the task, regardless of the chosen effort model and distance term. Adding the bonus term into the reward function improves performance significantly. However, even with completion bonus, models that lack the distance component are unable to achieve a success rate higher than 25% (only tested for EJK and zero effort). Among the considered three distance models, the exponential distance term demonstrates the highest success rates for most conditions. An exception is observed with the CTC effort model, where the absolute distance term performs considerably better than squared and exponential distance rewards.

We also analyse the time the model needs to press the button. Figure 3 (right) demonstrates that the completion times vary between the choice of the effort model and the distance. The zero effort model is not always the fastest, even when considering 100% success rates. Instead, the results depend on the chosen distance function.

5 Discussion

5.1 Discussion of Results and Guidelines

The results from our simulation study demonstrate that the task completion bonus is essential; all models trained without this bonus consistently failed to complete the task. They usually adopted a strategy of touching the buttons from the side. While this strategy maximizes the distance reward, it does not result in early termination of the episode, thus leading to a suboptimal total episode return. We thus suggest:

- G1 Include a task-specific completion bonus as the fundamental element of the reward function.

We also demonstrate that for the choice reaction task, it is not sufficient to choose a reward function consisting of solely sparse rewards. Adding an effort term did not improve performance, whereas including a distance reward term to guide the agent led to task completion when combined with a bonus term. While without an effort model, the absolute distance works best, in general, the precise choice of the distance function for best performance depends on the effort model. For the CTC effort model combined with a bonus term, the absolute distance achieves superior performance. We noticed that the absolute distance reward exhibits higher values at the beginning of the movement when compared to the other distance terms. Since with other distances the agent does not move towards the new button but instead rests on the previous one, we suspect

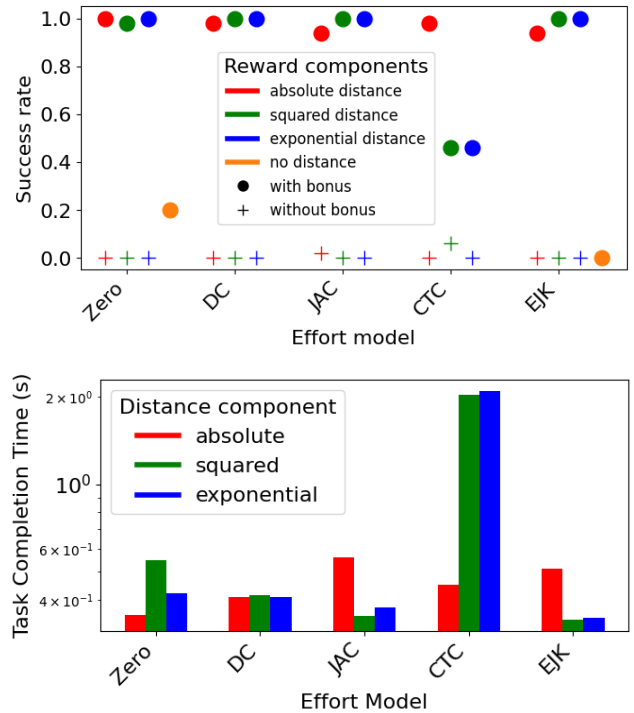


Figure 3: Success rates (top) and average task completion times (bottom) of models trained with different reward functions of type (1). Full parameter details are given in Table 1 in the appendix. Orange circles correspond to reward functions without distance rewards and with different bonus values (1, 8, and 50, all leading to the same success rate for a given effort model). The bottom figure shows the average task completion times of all models with completion bonus and a success rate of at least 50%. If a model does not manage to press a button within the time limit, the maximum time of four seconds is taken.

that the absolute distance provided "just enough" incentive. We suggest:

- G2 For complex scenarios involving vision or challenging tasks, integrate additional guidance components, such as distance-based terms, to guide the agent towards task completion.

In addition, our results suggest that an effort term is not necessary to generate successful movement trajectories. It is important to note that humans are capable of performing the task with different body poses. Consequently, it is possible that while the outcome remains constant, there is variation in movement pattern [18]. This problem can be solved by including different effort terms [18]. However, we did not identify any unreasonable patterns in the absence of the effort term. Our observations confirm earlier findings from [14], where a torque-actuated model was successfully trained to point in mid-air using a completion bonus only. The predicted movements in [14] even exhibit well-established movement characteristics such as Fitts' Law and the 2/3 Power Law. However, this was mainly attributed to an adaptive target size curriculum that effectively used state exploration to “guide” the RL agent towards reasonable target regions. It will be interesting to further explore to which extent this effect can be achieved through appropriate dense reward terms instead of curriculum learning. Since our qualitative and quantitative analysis did not indicate that implausible behaviour emerges without effort terms, based on the findings from our simulation study (see Figure 3), we suggest:

G3 Try without effort terms first and add one if instabilities occur. When adding effort terms, ensure comparability across different guidance components by normalizing their values and adjusting their weights where needed.

If an effort term is included, its weight can have a decisive effect on simulation quality. Excessively large effort weights prevent the model from moving at all, as the effort cost outweighs the incentive to act. Reducing effort weights incrementally enables progressive task completion in our experiments, from pressing a single button to hitting multiple targets. Conversely, assigning an effort weight that is too low diminishes its influence on the model's movements, possibly leading to an increase in unstable movements. We thus suggest:

G4 Adjust effort weights dynamically—decrease them if the model struggles to complete the task; increase them if movement instability is observed.

5.2 Limitations and Future Work

While our findings provide novel insights into the intricacies of reward functions and their effects on RL-based simulated users, our work is subject to several limitations. Our study focuses on a single interaction task, namely choice reaction, and a single biomechanical model. Consequently, future work should consider additional HCI tasks, such as pointing, tracking or keyboard typing [21, 23], and analyse the robustness of the considered and proposed reward functions to changes in the environment, task, and user model. This analysis could benefit from additional metrics, such as the total work done, and statements on robustness could be underpinned with statistical tests.

While the choice and number of trained policies (60) yielded valuable insights regarding plausibility (RQ.1), we only offer preliminary insights into sensitivities (RQ.2). We find that weights can be too large or too small, i.e., a sweetspot is needed. Additional training is required to provide more elaborate guidance on how to find that sweetspot, especially reward functions with multiple tunable weights.

Moreover, a more profound understanding of potential biases introduced by “shaping” reward terms, e.g., towards certain strategies such as extending the arm early during the movement, is needed. On a technical level, separating between sensory perception and motor control in the neural network structure instead of learning visuomotor using a single neural network (i.e., end-to-end) could enhance further analysis of RL-based biomechanical simulations.

Finally, the relation between reward function tuning (e.g., adding guidance costs) and established techniques to enhance the RL training process, such as adaptive automated curriculums [14, 16] or muscle-specific state exploration techniques [3, 11, 39], are open questions. While more evidence is needed, our initial analysis suggests that distance reward components have the potential to restrict the motor control space to biomechanically plausible regions.

6 Conclusion

Reward function design plays a crucial role for RL-based biomechanical simulations. Using a choice reaction task as a test-bed, we have analysed the individual and combined effects of three essential reward function components, namely task completion, target proximity, and effort terms. Our simulation study reveals that a combination of sparse completion bonus and dense proximity rewards is essential for task success. Interestingly, effort terms are dispensable if appropriate proximity rewards are used; otherwise, they need to be carefully weighted. Our work emphasises the need for a better understanding of the subtleties involved in training musculoskeletal models, for a variety of interaction tasks. By providing guidelines and first principles for reward function design, this work contributes towards the use of RL-based user simulations as a practical tool for HCI research and design.

Acknowledgments

This work was supported by EPSRC grant EP/W02456X/1. Hannah Selder and Arthur Fleig acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the programme Center of Excellence for AI-research „Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig“, project identification number: ScaDS.AI.

References

- [1] M. M. Ayoub, Arun Walvekar, and Mike Petruno. 1974. A Biomechanical Model for the Upper Extremity Using Optimization Techniques. *SAE Transactions* 83 (1974), 1139–1148. <http://www.jstor.org/stable/44734426>
- [2] Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, Jürgen Steimle, and Tino Weinkauff. 2015. Performance and ergonomics of touch surfaces: A comparative study using biomechanical simulation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1817–1826.
- [3] Cameron Berg, Vittorio Caggiano, and Vikash Kumar. 2024. Sar: Generalization of physiological agility and dexterity via synergistic action representation. *Autonomous Robots* 48, 8 (2024), 28.
- [4] Bastien Berret, Enrico Chiovetto, Francesco Nori, and Thierry Pozzo. 2011. Evidence for Composite Cost Functions in Arm Movement Planning: An Inverse Optimal Control Approach. *PLoS Computational Biology* 7, 10 (Oct. 2011), e1002183. doi:10.1371/journal.pcbi.1002183
- [5] Vittorio Caggiano, Sudeep Dasari, and Vikash Kumar. 2023. MyoDex: a generalizable prior for dexterous manipulation. In *Proceedings of the 40th International Conference on Machine Learning (Honolulu, Hawaii, USA) (ICML '23)*. JMLR.org, New York, NY, USA, 20 pages.

- [6] Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. 2022. MyoSuite: A Contact-rich Simulation Suite for Musculoskeletal Motor Control. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference (Proceedings of Machine Learning Research, Vol. 168)*, Roya Firoozi, Negar Mehr, Esen Yel, Rika Antonova, Jeannette Bohg, Mac Schwager, and Mykel Kochenderfer (Eds.). PMLR, New York, NY, USA, 492–507. <https://proceedings.mlr.press/v168/caggiano22a.html>
- [7] Jhon P.F. Charaja, Isabell Wochner, Pierre Schumacher, Winfried Ilg, Martin Giese, Christophe Maufroy, Andreas Bulling, Syn Schmitt, Georg Martius, and Daniel F.B. Haeufle. 2024. Generating Realistic Arm Movements in Reinforcement Learning: A Quantitative Comparison of Reward Terms and Task Requirements. In *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, Piscataway, NJ, USA, 562–568. doi:10.1109/biorob60516.2024.10719719
- [8] Noshaba Cheema, Laura A. Frey-Law, Kourosh Naderi, Jaakko Lehtinen, Philipp Slusallek, and Perttu Hämmäläinen. 2020. Predicting Mid-Air Interaction Movements and Fatigue Using Deep Reinforcement Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–13. doi:10.1145/3313831.3376701
- [9] Xiuli Chen, Gilles Bailly, Duncan P. Brumby, Antti Oulasvirta, and Andrew Howes. 2015. The Emergence of Interactive Behavior: A Model of Rational Menu Search. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 4217–4226. doi:10.1145/2702123.2702483
- [10] Xiuli Chen, Sandra Dorothee Starke, Chris Baber, and Andrew Howes. 2017. A Cognitive Model of How People Make Decisions Through Interaction with Visual Displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, Denver Colorado USA, 1205–1216. doi:10.1145/3025453.3025596
- [11] Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Zixiang Huang, and Alexander Mathis. 2023. Latent exploration for reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (New Orleans, LA, USA) (NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 2466, 23 pages.
- [12] Michael Damsgaard, John Rasmussen, Søren Torholm Christensen, Egidijus Surma, and Mark De Zee. 2006. Analysis of musculoskeletal systems in the AnyBody Modeling System. *Simulation Modelling Practice and Theory* 14, 8 (Nov. 2006), 1100–1111. doi:10.1016/j.simpat.2006.09.001 Publisher: Elsevier BV.
- [13] Scott L. Delp, Frank C. Anderson, Allison S. Arnold, Peter Loan, Ayman Habib, Chand T. John, Eran Guendelman, and Darryl G. Thelen. 2007. OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. *IEEE Transactions on Biomedical Engineering* 54, 11 (2007), 1940–1950. doi:10.1109/TBME.2007.901024
- [14] Florian Fischer, Miroslav Bachinski, Markus Klar, Arthur Fleig, and Jörg Müller. 2021. Reinforcement learning control of a biomechanical model of the upper extremity. *Scientific Reports* 11, 1 (2021), 14445.
- [15] Florian Fischer, Arthur Fleig, Markus Klar, and Jörg Müller. 2022. Optimal Feedback Control for Modeling Human–Computer Interaction. *ACM Transactions on Computer-Human Interaction* 29, 6 (Dec. 2022), 1–70. doi:10.1145/3524122
- [16] Florian Fischer, Aleksii Ikkala, Markus Klar, Arthur Fleig, Miroslav Bachinski, Roderick Murray-Smith, Perttu Hämmäläinen, Antti Oulasvirta, and Jörg Müller. 2024. SIM2VR: Towards Automated Biomechanical Testing in VR. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. ACM, Pittsburgh PA USA, 1–15. doi:10.1145/3654777.3676452
- [17] T Flash and N Hogan. 1985. The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience* 5, 7 (July 1985), 1688–1703. doi:10.1523/JNEUROSCI.05-07-01688.1985
- [18] Emmanuel Guigon, Pierre Baraduc, and Michel Desmurget. 2007. Computational Motor Control: Redundancy and Invariance. *Journal of Neurophysiology* 97, 1 (Jan. 2007), 331–347. doi:10.1152/jn.00290.2006
- [19] Yixu He, Yang Liu, Lan Yang, and Xiaobo Qu. 2024. Exploring the design of reward functions in deep reinforcement learning-based vehicle velocity control algorithms. *Transportation Letters* 16, 10 (Nov. 2024), 1338–1352. doi:10.1080/19427867.2024.2305018
- [20] Lorenz Hetzel, John Dudley, Anna Maria Feit, and Per Ola Kristensson. 2021. Complex interaction as emergent behaviour: Simulating mid-air virtual keyboard typing using reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics* 27, 11 (2021), 4140–4149.
- [21] Lorenz Hetzel, John Dudley, Anna Maria Feit, and Per Ola Kristensson. 2021. Complex Interaction as Emergent Behaviour: Simulating Mid-Air Virtual Keyboard Typing using Reinforcement Learning. *IEEE Transactions on Visualization and Computer Graphics* 27, 11 (2021), 4140–4149. doi:10.1109/TVCG.2021.3106494
- [22] Seokhyun Hwang, Jeongseok Oh, Seongjun Kang, Minwoo Seong, Ahmed Ibrahim Ahmed Mohamed Elsharkawy, and Seungjun Kim. 2024. ErgoPulse: Electrifying Your Lower Body With Biomechanical Simulation-based Electrical Muscle Stimulation Haptic System in Virtual Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–21. doi:10.1145/3613904.3642008
- [23] Aleksii Ikkala, Florian Fischer, Markus Klar, Miroslav Bachinski, Arthur Fleig, Andrew Howes, Perttu Hämmäläinen, Jörg Müller, Roderick Murray-Smith, and Antti Oulasvirta. 2022. Breathing Life Into Biomechanical User Models. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. ACM, Bend OR USA, 1–14. doi:10.1145/3526113.3545689
- [24] Janghyeon Kim, Ho-Jin Jung, Dae Han Sim, Ji-Hyeon Yoo, Song Woo Kim, and Han Ul Yoon. 2023. An Approach to Design a Biomechanically-Inspired Reward Function to Solve a Patience Cube Under Reinforcement Learning Framework. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Piscataway, NJ, USA, 5653–5660. doi:10.1109/IROS55552.2023.10341831
- [25] Markus Klar, Florian Fischer, Arthur Fleig, Miroslav Bachinski, and Jörg Müller. 2023. Simulating Interaction Movements via Model Predictive Control. *ACM Transactions on Computer-Human Interaction* 30, 3 (June 2023), 1–50. doi:10.1145/3577016
- [26] Ariel Kwiatkowski, Vicky Kalogeiton, Julien Pettré, and Marie-Paule Cani. 2023. Reward Function Design for Crowd Simulation via Reinforcement Learning. In *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction and Games (Rennes, France) (MIG '23)*. Association for Computing Machinery, New York, NY, USA, Article 4, 7 pages. doi:10.1145/3623264.3624452
- [27] Dan Liu and Emanuel Todorov. 2007. Evidence for the flexible sensorimotor strategies predicted by optimal feedback control. *Journal of Neuroscience* 27, 35 (2007), 9354–9368.
- [28] Pascal Madeleine, Afshin Samani, Mark de Zee, and Uwe Kersting. 2011. Biomechanics of Human Movement. In *IFMBE Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, 237–240. doi:10.1007/978-3-642-21683-1_60 ISSN: 1680-0737, 1433-9277.
- [29] J. Alberto Álvarez Martín, Henrik Gollee, Jörg Müller, and Roderick Murray-Smith. 2021. Intermittent Control as a Model of Mouse Movements. *ACM Transactions on Computer-Human Interaction* 28, 5 (Oct. 2021), 1–46. doi:10.1145/3461836 Publisher: Association for Computing Machinery (ACM).
- [30] Matthew Millard, Anna Lena Emonds, Monika Harant, and Katja Mombaur. 2019. A reduced muscle model and planar musculoskeletal model fit for the simulation of whole-body movements. *Journal of biomechanics* 89 (2019), 11–20.
- [31] Roderick Murray-Smith, Antti Oulasvirta, Andrew Howes, Jörg Müller, Aleksii Ikkala, Miroslav Bachinski, Arthur Fleig, Florian Fischer, and Markus Klar. 2022. What simulation can do for HCI research. *Interactions* 29, 6 (Nov. 2022), 48–53. doi:10.1145/3564038
- [32] Arne J Nagengast, Daniel A Braun, and Daniel M Wolpert. 2011. Risk sensitivity in a motor task with speed-accuracy trade-off. *Journal of neurophysiology* 105, 6 (2011), 2668–2674.
- [33] Jorge Nocedal and Stephen J. Wright. 2006. *Numerical optimization* (2. ed. ed.). Springer, New York, NY. http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+502988711&sourceid=fwb_bibsonomy
- [34] Katharine Nowakowski, Philippe Carvalho, Jean-Baptiste Six, Yann Maillet, Anh Tu Nguyen, Ismail Seghiri, Loick M’Pemba, Theo Marcille, Sy Toan Ngo, and Tien-Tuan Dao. 2021. Human locomotion with reinforcement learning using bioinspired reward reshaping strategies. *Medical & Biological Engineering & Computing* 59, 1 (Jan. 2021), 243–256. doi:10.1007/s11517-020-02309-3
- [35] Tommaso Proietti, Emmanuel Guigon, Agnès Roby-Brami, and Nathanaël Jarrassé. 2017. Modifying upper-limb inter-joint coordination in healthy subjects by training with a robotic exoskeleton. *Journal of NeuroEngineering and Rehabilitation* 14, 1 (Dec. 2017), 55. doi:10.1186/s12984-017-0254-x
- [36] Ivo Roupá, Mariana Rodrigues da Silva, Filipe Marques, Sérgio B Gonçalves, Paulo Flores, and Miguel Tavares da Silva. 2022. On the modeling of biomechanical systems for human movement analysis: a narrative review. *Archives of Computational Methods in Engineering* 29, 7 (2022), 4915–4958.
- [37] Isabel C.N. Sacco, Francis Trombini-Souza, and Eneida Yuri Suda. 2023. Impact of biomechanics on therapeutic interventions and rehabilitation for major chronic musculoskeletal conditions: A 50-year perspective. *Journal of Biomechanics* 154 (2023), 111604. doi:10.1016/j.jbiomech.2023.111604
- [38] Katherine R Saul, Xiao Hu, Craig M Goehler, Meghan E Vidt, Melissa Daly, Anca Velisar, and Wendy M Murray. 2015. Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Computer methods in biomechanics and biomedical engineering* 18, 13 (2015), 1445–1458.
- [39] Pierre Schumacher, Daniel F.B. Haeufle, Dieter Büchler, Syn Schmitt, and Georg Martius. 2023. DEP-RL: Embodied Exploration for Reinforcement Learning in Overactuated and Musculoskeletal Systems. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*. ICLR, Appleton, WI, USA, 1–29. https://openreview.net/forum?id=C-xa_D3oTj6
- [40] Seungmoon Song, Łukasz Kidziński, Xue Bin Peng, Carmichael Ong, Jennifer Hicks, Sergey Levine, Christopher G Atkeson, and Scott L Delp. 2021. Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation. *Journal of neuroengineering and rehabilitation* 18 (2021), 1–17.
- [41] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Piscataway NJ USA, 5026–5033. doi:10.1109/IROS.2012.6386109
- [42] Emanuel Todorov and Michael I. Jordan. 2002. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience* 5, 11 (Nov. 2002), 1226–1235.

- doi:10.1038/n963 Publisher: Springer Science and Business Media LLC.
- [43] Yasuhiro Wada, Yuichi Kaneko, Eri Nakano, Rieko Osu, and Mitsuo Kawato. 2001. Quantitative examinations for multi joint arm trajectory planning—using a robust calculation algorithm of the minimum commanded torque change trajectory. *Neural Networks* 14, 4-5 (May 2001), 381–393. doi:10.1016/S0893-6080(01)00026-0
- [44] Bingjun Wan and Gongbing Shan. 2016. Biomechanical modeling as a practical tool for predicting injury risk related to repetitive muscle lengthening during learning and training of human complex motor skills. *SpringerPlus* 5 (2016), 1–13.
- [45] Rubin Wang and Fanji Gu (Eds.). 2011. *Advances in Cognitive Neurodynamics (II): Proceedings of the Second International Conference on Cognitive Neurodynamics - 2009*. Springer Netherlands, Dordrecht. doi:10.1007/978-90-481-9695-1
- [46] DA Winter. 1984. Biomechanics of human movement with applications to the study of human locomotion. *Critical reviews in biomedical engineering* 9, 4 (1984), 287–314. <http://europepmc.org/abstract/MED/6368126>
- [47] Isabell Wochner, Danny Driess, Heiko Zimmermann, Daniel F. B. Haeufle, Marc Toussaint, and Syn Schmitt. 2020. Optimality Principles in Human Point-to-Manifold Reaching Accounting for Muscle Dynamics. *Frontiers in Computational Neuroscience* 14 (May 2020), 38. doi:10.3389/fncom.2020.00038
- [48] Yangmengfei Xu, Vincent Crocher, Justin Fong, Ying Tan, and Denny Oetomo. 2021. Inducing Human Motor Adaptation Without Explicit Error Feedback: A

Motor Cost Approach. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), 1403–1412. doi:10.1109/TNSRE.2021.3096516

A Appendix

The following tables present the reward functions used to train the policies. We recall from (1) that a reward function consists of the bonus, distance, and effort term and therefore amounts to

$$r_t = w_{\text{bonus}} \cdot f_{\text{bonus}}(\cdot) - w_{\text{distance}} \cdot f_{\text{distance}}(\cdot) - w_{\text{effort}} \cdot f_{\text{effort}}(\cdot).$$

The coefficients of the effort models are based on the results in [7, 25]. For instance, the reward function of the run with ID 1 is:

$$r_t = \begin{cases} 8 - \frac{r_{\text{energy}} + 8 \cdot r_{\text{jer}} + r_{\text{work}}}{10}, & \text{correct button pressed,} \\ 0 - \frac{1 - e^{-10 \cdot \text{dist}}}{10} - \frac{r_{\text{energy}} + 8 \cdot r_{\text{jer}} + r_{\text{work}}}{10}, & \text{else.} \end{cases}$$

Table 1: Parameters for trained policies with distance weight $w_{\text{distance}} = 1$ and bonus weight $w_{\text{bonus}} = 1$.

ID	Effort	w_{effort}	Effort coefficients	Distance	Bonus b
1	EJK	0.8	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	8
2	EJK	0.8	$c_1=1, c_2=8, c_3=1$	D_{absolute}	8
3	EJK	0.8	$c_1=1, c_2=8, c_3=1$	D_{squared}	8
4	EJK	0.8	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	0
5	EJK	0.8	$c_1=1, c_2=8, c_3=1$	D_{absolute}	0
6	EJK	0.8	$c_1=1, c_2=8, c_3=1$	D_{squared}	0
7	JAC	0.1	$c_1=0.0198, c_2=6.67 \cdot 10^{-5}$	$D_{\text{exponential}}$	8
8	JAC	0.1	$c_1=0.0198, c_2=6.67 \cdot 10^{-5}$	D_{absolute}	8
9	JAC	0.1	$c_1=0.0198, c_2=6.67 \cdot 10^{-5}$	D_{squared}	8
10	JAC	0.1	$c_1=0.0198, c_2=6.67 \cdot 10^{-5}$	$D_{\text{exponential}}$	0
11	JAC	0.1	$c_1=0.0198, c_2=6.67 \cdot 10^{-5}$	D_{absolute}	0
12	JAC	0.1	$c_1=0.0198, c_2=6.67 \cdot 10^{-5}$	D_{squared}	0
13	CTC	0.01	$c_1=0.649, c_2=0.0177$	$D_{\text{exponential}}$	8
14	CTC	0.01	$c_1=0.649, c_2=0.0177$	D_{absolute}	8
15	CTC	0.01	$c_1=0.649, c_2=0.0177$	D_{squared}	8
16	CTC	0.01	$c_1=0.649, c_2=0.0177$	$D_{\text{exponential}}$	0
17	CTC	0.01	$c_1=0.649, c_2=0.0177$	D_{absolute}	0
18	CTC	0.01	$c_1=0.649, c_2=0.0177$	D_{squared}	0
19	DC	0.01	$c_1=0.1477$	$D_{\text{exponential}}$	8
20	DC	0.01	$c_1=0.1477$	D_{absolute}	8
21	DC	0.01	$c_1=0.1477$	D_{squared}	8
22	DC	0.01	$c_1=0.1477$	$D_{\text{exponential}}$	0
23	DC	0.01	$c_1=0.1477$	D_{absolute}	0
24	DC	0.01	$c_1=0.1477$	D_{squared}	0
25	Zero	0		$D_{\text{exponential}}$	8
26	Zero	0		D_{absolute}	8
27	Zero	0		D_{squared}	8
28	Zero	0		$D_{\text{exponential}}$	0
29	Zero	0		D_{absolute}	0
30	Zero	0		D_{squared}	0
31	Zero	0			8
32	Zero	0			50
33	EJK	0.8	$c_1=1, c_2=8, c_3=1$		1
34	EJK	0.8	$c_1=1, c_2=8, c_3=1$		8
35	EJK	0.8	$c_1=1, c_2=8, c_3=1$		50

Table 2: Further trainings with exponential distance, distance weight $w_{\text{distance}} = 1$, and bonus weight $w_{\text{bonus}} = 1$

ID	Effort	w_{effort}	Effort coefficients	Distance	Bonus b
36	EJK	16	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	8
37	EJK	8	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	8
38	EJK	4	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	8
39	EJK	1.6	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	8
40	EJK	0.4	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	8
41	EJK	0.16	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	8
42	EJK	0.08	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	8
43	EJK	0.04	$c_1=1, c_2=8, c_3=1$	$D_{\text{exponential}}$	8
44	JAC	1	$c_1=0.0198, c_2=6.67 \cdot 10^{-5}$	$D_{\text{exponential}}$	8
45	JAC	1	$c_1=0.0198, c_2=6.67 \cdot 10^{-5}$	$D_{\text{exponential}}$	0
46	JAC	0.01	$c_1=0.0198, c_2=6.67 \cdot 10^{-5}$	$D_{\text{exponential}}$	8
47	JAC	0.01	$c_1=0.0198, c_2=6.67 \cdot 10^{-4}$	$D_{\text{exponential}}$	8
48	CTC	1	$c_1=0.649, c_2=0.0177$	$D_{\text{exponential}}$	8
49	CTC	1	$c_1=0.649, c_2=0.0177$	$D_{\text{exponential}}$	0
50	CTC	0.1	$c_1=0.649, c_2=0.0177$	$D_{\text{exponential}}$	8
51	CTC	0.001	$c_1=0.649, c_2=0.0177$	$D_{\text{exponential}}$	8
52	DC	1	$c_1=0.1477$	$D_{\text{exponential}}$	8
53	DC	1	$c_1=0.1477$	$D_{\text{exponential}}$	0
54	DC	0.001	$c_1=0.1477$	$D_{\text{exponential}}$	8
55	DC	1	$c_1=0.0001$	$D_{\text{exponential}}$	8
56	DC	1	$c_1=0.0001$	$D_{\text{exponential}}$	8
57	DC	5	$c_1=0.0001$	$D_{\text{exponential}}$	8
58	DC	10	$c_1=0.0001$	$D_{\text{exponential}}$	8
59	DC	50	$c_1=0.0001$	$D_{\text{exponential}}$	8
60	DC	100	$c_1=0.0001$	$D_{\text{exponential}}$	8