Continuous Learning of Ordinal User Preferences on Wearable Devices

Simón Weinberger[1,2] (✉), Jairo Cugliari[2], and Aurélie Le Cain[1]

[1] Essilor International, affiliate of EssilorLuxottica, 39 Bd. Jean Baptiste Oudry, 94000 Créteil weinbes@essilor.fr
[2] Laboratoire ERIC, 5 Av. Pierre Mendès France, 69500 Bron
jairo.cugliari@univ-lyon2.fr

Abstract. Wearable devices allow collecting data at an individual level, which can be used to propose an unseen degree of personalization for a broad domain of applications. For instance, we focus on electrochromic frames that allow to manually change the lens' tint, or automatically, based on an ambient light sensor. We aim to use the user's interactions with his frame to adapt this automatic mode to better consider his preferences. From a technical standpoint, this is a difficult task, as prediction and estimation cannot be done separately. That is why we approach this industrial problem from a reinforcement learning perspective: a policy must control the tint class in such a way that the number of user interactions is minimized. A particularity of this problem is that there is an inherent notion of order between the finite proposed tint classes, as some are darker than others. The usual Boltzmann parametrization does not account for this. Thus, we develop and implement policy gradient methods for ordinal policies. Using a simulation setting, we show that ignoring the ordinal structure of the response variables yields a suboptimal strategy. Additionally, we tested this technique with real users in controlled conditions; as the tint-control mode updated, the number of user interactions decreased. At last, using ordinal policies can be adapted to a deep reinforcement learning context, solving classic problems with continuous actions using discretization of this space.

Keywords: Reinforcement Learning · On-policy Policy Gradient Methods · Ordinal regression · Wearables.

1 Introduction

Wearable devices have revolutionized the way we interact with technology, offering personalized experiences tailored to individual preferences and needs [16]. From tracking fitness goals [8] to monitoring health metrics [13], wearables have become indispensable companions in our daily lives. However, the dynamic nature of user preferences and habits presents a challenge in maintaining optimal performance over time. As users' needs evolve, the effectiveness of pre-defined models and algorithms may diminish, highlighting the necessity for continuous learning mechanisms.

Traditional machine learning approaches often struggle to adapt to the changing preferences observed in wearable device users; moreover, the inherent constraints of wearable devices, such as limited computational resources and battery life, further exacerbate this challenge [17]. In industrial applications, where reliability and robustness are paramount, the need to fortify decision-making processes becomes even more pronounced.

In this research, we propose leveraging Reinforcement Learning (RL) techniques to facilitate continuous learning of ordinal user preferences on wearable devices. By formulating the learning problem as one of policy optimization, we aim to capture the nuanced ranking of user preferences. To accommodate the constraints of wearable devices, we integrate a generalized vectorial linear model that accounts for resource limitations and computational efficiency.

Additionally, our proposed framework can also be used in a Deep Reinforcement Learning (DRL) context to solve classical continuous action RL problems by discretizing the set of actions into a finite and ordered set of actions. Doing so yields similar results than using continuous actions.

1.1 A general learning setting for ordinal user preferences

Let us consider a system that pilots automatically a setting of a wearable device, using the wearable's sensors. In addition, let us suppose that there are $K \in \mathbb{N}$ settings and there exists an order relationship between these categories. The particular setting will be called level. Additionally, the user may interact with the wearable and manually adjust the setting at any moment.

Ideally, the system pilots the setting in a way that provides the best user experience. As a measure of user experience, it is reasonable to use the number of interactions or the eventual difference between the system-chosen and user-chosen levels. Indeed, if the system is well adapted for the user, the user should not manually choose a level often, and inversely if the system is ill-adapted

We formulate this problem in a RL setting. The state space $\mathcal{S}$ corresponds to the measures of the wearable device, the action space $\mathcal{A}$ corresponds to the possible setting levels and the reward space $\mathcal{R}$ is either a penalty if the user manually changed the level or the eventual absolute value difference between the system-proposed and user-chosen levels.

The data collection process would be as follows: the wearable's sensors collect a measure *(state)*. Based on that measure, the system proposes a new level to the user *(action)*. Then the user would either accept the proposed level or choose another level *(reward)*. The objective is to have a system that proposes levels that are often accepted by the user or would be close to the levels the user prefers. This process would be repeated at each instant for a certain amount of time (for example, a day), creating an episode $\boldsymbol{\tau}$ of length $T \in \mathbb{N}$,

$$\boldsymbol{\tau} = (s_t, a_t, r_t)_{t=0}^{T} \quad ; \quad s_t \in \mathcal{S},\ a_t \in \mathcal{A},\ r_t \in \mathcal{R}.$$

As in standard RL [21], we suppose the state-reward transitions are governed by the distribution defined by

$$p(s', r|s, a) = \mathbb{P}(S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a).$$

And episodes verify the Markov hypothesis:

$$\mathbb{P}(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a, \dots, S_0, A_0) = p(s', r | s, a).$$

The tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$ is a Markov Decision Process (MDP) [21]. Let us note $\mathcal{P}(\mathcal{A})$ the set of probability distributions over the action space $\mathcal{A}$, at each step, actions are taken by according to a policy: $\pi(\cdot|s_t) \in \mathcal{P}(\mathcal{A})$. For a given discount factor $\gamma \in [0, 1[$, action $a \in \mathcal{A}$ and state $s \in \mathcal{S}$ and a policy π, we define the state-action value function, $Q^\pi(s, a)$ and state value function $V^\pi(s)$:

$$Q^\pi(s,a) = \mathbb{E}_\pi \left(\sum_{t=0}^{T} \gamma^t R_t | S_0 = s, A_0 = a \right) ; \; V^\pi(s) = \mathbb{E}_\pi \left(\sum_{t=0}^{T} \gamma^t R_t | S_0 = s \right).$$

Additionally, we suppose that the policy belongs to a given parametric space:

$$\pi \in \left\{ \pi_\theta : \mathcal{S} \to \mathcal{P}(\mathcal{A}) | \theta \in \Theta \subset \mathbb{R}^d \right\}.$$

In this article, the policy families of interest are the Boltzmann distribution, also known as softmax or multinomial distribution, and policies defined by the cumulative ordinal model. Both of these allow a policy to choose a level, the first family does not consider the order between levels, the latter does. Let ν be any initial state distribution, the goal of policy optimization is to find the maximum of the objective function $J(\cdot)$:

$$J(\theta) = \mathop{\mathbb{E}}_{\substack{a_t \sim \pi_\theta(\cdot|s_t) \\ s_0 \sim \nu}} \left(\sum_{t=0}^{T} \gamma^t R_t \right) \quad ; \quad \theta \in \Theta.$$

It is worth noting that the problem of parametrizing policies is orthogonal to the method used for policy optimization. Most policy-based methods, or actor-critic methods, can be adapted for different types of policies.

1.2 Electrochromic adaptative automatic mode

In our industrial application, we chose to work with smart lenses, which admits some kind of customization. EssilorLuxottica is developing a smart eyewear equipped with electrochromic lenses. The tint of this eyewear can change by passing an electric signal, allowing the user to choose one tint among four: C_0, C_1, C_2, C_3. There is an inherent order relationship between these classes, as some are clearer than others. There exists an automatic mode for these frames that controls the tint by using measures from an Ambient Light Sensor (ALS) and comparing these values to three predefined, ordered thresholds $\rho_1 < \rho_2 < \rho_3$. If the ALS measure is below ρ_1, the clearest tint (C_0) is chosen, if the value lays between ρ_1 and ρ_2, C_1 is chosen, if the value is between ρ_2 and ρ_3, C_2 is chosen and if the value is greater than ρ_3 the darkest tint (C_3) is chosen. This tint control could be used to create a *hybrid mode*: the tint is controlled as previously described, except when the user manually changes the frame's tint, when this

happens the automatic control is disabled for a certain amount of time or until the situation changes.

In previous work, we studied how to personalize the tint control mode using labeled data issued from the manual usage of these smart lenses, adopting a supervised approach [26]. In contrast, this article is about how to personalize this hybrid mode so that the user's preferences are considered using observations collected while in hybrid mode. We do so applying the RL paradigm, as described in Section 1.1.

For this particular problem, we use only the ALS measures, and because the eye reacts to light on a logarithmic scale [24], we consider the state space as the $\log 10$ of the ALS measures (which are between 0 and 5). The set of actions is the set of tint classes. The reward is -1 if there was a user interaction and 0 otherwise. Concretely:

$$\mathcal{S} = [0, 5] \ ; \ \mathcal{A} = \{C_0, C_1, C_2, C_3\} \ ; \ \mathcal{R} = \{-1, 0\}.$$

1.3 Main Contributions

We introduce a family of policies taking actions on a finite, totally ordered action space. These policies are an adaptation of linear ordinal regression models [3] and CORAL neural networks [4] into a RL setting.

We provide necessary conditions assuring that the policy has a form of smoothness. We reparametrize the parameters of this model, allowing to optimize in an unconstrained space. This is done in a way that guarantees the smoothness of the policy and respects constraints over the parameters of the policy. We use this to implement policy optimization using REINFORCE, NPG, TRPO and PPO for this policy.

We demonstrate that, in certain scenarios where there is a notion of order among actions, using an ordinal policy converges faster to better policies than using a softmax distribution (Section 3.1). This suggests that considering the notion of order among action can be beneficial for tackling real-world problems.

We tested this method with real electrochromic prototypes worn by real users, in a controlled setting (Section 3.2). The results of this study are positive: as the tint control parameters are updated, the number of user interactions diminishes.

We show that using an ordinal policy instead of a continuous action policy can perform as well as a traditional continuous action policy in standard RL benchmark environments (Section 3.3).

1.4 Related work

In a supervised context, ordinal regression models were introduced by McCullagh [15]; those models belong to the more general family of Vector Generalized Additive Models (VGAM), which were introduced by Yee and Wild [28]. This approach can be extended in even a more general setting: the predictor can, in fact, be parametrized by a neural network. Indeed, ordinal regression can be

achieved using binary extended classification, as explained by Li and Lin [14], allowing to perform ordinal regression using any binary classification algorithm. This was used to implement COnsistent RAnk Logits (CORAL) in neural networks [4] and Conditional Ordinal Regression for neural networks (CORN) [20]. Both methods augment the training data set to encode an ordinal regression problem as a binary classification problem, and then train a neural network using a particular loss function that assures that the obtained rank logits are ordered.

In a RL context, the parametrization of policies taking action or ordered sets has received little to no attention. Although there are some articles that tackle solving continuous actions by using discretization, for example, this was done by Seyde *et al.* [19] and Tang and Agrawal [22] to solve environments with continuous actions, which achieved state-of-the-art convergence rates in different benchmark environments. The first article used a dichotomous action space, using higher and lower actions as actions; the latter used a discretization with more classes by implementing a policy inspired by "stick-breaking" [12]. Both approaches rely on defining neural networks outputting logits to take actions, the latter article then accordingly defines a notion of order between classes (Equation 4 of [22]). Although not directly linked to ordinal actions, efforts have been made to consider policies taking actions on structured continuous actions spaces, as presented by Wu *et al.* [27], or considering monotonic policies as explained by Feng *et al.* [6], relying on monotonic neural networks [25]. Either way, using monotonic or structured policies, yielded more stable numerical results.

We propose using a different parametrization over ordinal actions, based on a thresholded model [14], which is a latent variable model relying on a scalar predictor and ordered thresholds. Unlike the method proposed by Tang and Agrawal [22], our parametrization enforces the notion of order among actions, which is desirable for our industrial application. For instance, if the state space is defined by ALS measurements, the scalar predictor is proportional to these. As a result, the predicted classes will follow an ordered structure: if the proportionality coefficient is positive, an increase in ALS values will correspond to the prediction of progressively darker classes.

2 Theory: Ordinal Policies

For simplicity, in this section, we first present an ordinal policy for one-dimensional actions, present some results and finally extend it to multivariate ordinal actions, by factorizing across dimensions.

2.1 Definitions

We consider the situation where the action space has a finite number of elements $\mathcal{A} = \{a_k\}_{k=1}^{K}$, with K being a natural number greater or equal than three, and there exists an order relationship between actions:

$$a_1 < a_2 < \ldots < a_{K-1} < a_K. \tag{1}$$

For any natural numbers $a, b \in \mathbb{N}$, let us note $[\![a, b]\!]$ the set of whole numbers between a and b. Without losing generality, we suppose $\mathcal{A} = [\![1, K]\!]$. In the general discrete setting, the only way to parametrize a distribution is to use the discrete probability distribution, using a multinomial model over logits for each class, for example. In the ordinal context, there is another possible parametrization: using the cumulative distribution function, which is well-defined. Let us consider a function $g_\omega \colon \mathcal{S} \to \mathbb{R}$, parametrized by a weight, $\omega \in \Omega$, and consider $K-1$ ordered thresholds $(\tau_k)_{k=1}^{K-1}$:

$$\tau_1 < \tau_2 < \ldots < \tau_{K-2} < \tau_{K-1}.$$

Let $\sigma(x) = (1 + \exp(-x))^{-1}$ be the sigmoid function. A policy π over the set $\mathcal{A}$ is defined by the relation:

$$\sigma^{-1}\left(\mathbb{P}(A \leq j | S = s)\right) = \tau_j - g_\omega(s). \tag{2}$$

Indeed, using the convention $\tau_0 = -\infty$ and $\tau_K = \infty$, Equation (2) induces a probability distribution on the action space $\mathcal{A}$:

$$\pi(a|s) = \sigma(\tau_a - g_\omega(s)) - \sigma(\tau_{a-1} - g_\omega(s)) \quad ; \quad a \in \mathcal{A}.$$

It is worth noting that because the thresholds $(\tau_j)_{j=1}^{K-1}$ are ordered, the obtained cumulative probabilities are assured to be ordered:

$$\mathbb{P}(A \leq 0 | S = s) < \mathbb{P}(A \leq 1 | S = s) < \ldots < \mathbb{P}(A \leq K-1 | S = s).$$

Definition (2) is equivalent to the following latent variable relationship:

$$\begin{cases} A^* = g_\omega(s) + e \quad ; \quad e \sim \text{Logistic}(0, 1), \\ A = j \quad \Leftrightarrow \quad \tau_{j-1} < A^* \leq \tau_j. \end{cases} \tag{3}$$

This latter definition is more interpretable: the function g_ω is a map between the state space and $\mathbb{R}$, when it is high, the policy will often take high actions, when low, the policy will take low actions. Let us note $\Delta_{K-1} \subset \mathbb{R}^{K-1}$ the set of ordered thresholds:

$$\Delta_{K-1} = \left\{(x_1, \ldots, x_{K-1}) \in \mathbb{R}^{K-1} | x_{i+1} - x_i > 0, \quad \forall i \in [\![1, K-1]\!]\right\}.$$

Definition 1. *A policy π is said to be an ordinal policy if it verifies Equation (2). This parametric family is parametrized by $\Theta = \Omega \times \Delta_{K-1}$.*

It is worth noting that unlike normal and multinomial distributions, this ordinal distribution is not in the exponential family, and therefore it is not straightforward that usual policy improvement methods work with this ordinal policy.

Definition 2. *Let β be a positive real number and a function $f \colon \mathbb{R}^p \to \mathbb{R}$, f is said to be β-smooth, if its gradient is β-Lipschitz. Namely, $\forall a, b \in \mathbb{R}^p$:*

$$\|\nabla f(a) - \nabla f(b)\|_2 \leq \beta \|a - b\|_2.$$

In general, a suitable condition for gradient optimization techniques is that the function to optimize must be β-smooth. For instance, in the RL context, the improvement of policies with NPG, depends on the logarithm of the probability density function of the policy being β-smooth (Theorem 20 in [2]).

Definition 3. *A policy is said to be β-smooth if $\theta \mapsto \log \pi_\theta(.|s)$ is β-smooth for every $s \in \mathcal{S}$.*

It is straightforward to prove that a policy is β-smooth if it belongs in the exponential family. Yet, the introduced ordinal policy is neither in the exponential family nor β-smooth, even with a linear predictor. To have a β-smooth ordinal policy, it is necessary to assure that the distance between thresholds is sufficiently large. Let $\varepsilon > 0$, let us introduce the set $\Delta_{K-1}^{\varepsilon}$:

$$\Delta_n^{\varepsilon} = \{(x_1, \ldots, x_n) \in \mathbb{R}^n | x_{i+1} - x_i > \varepsilon, \, \forall i \in [\![1, n]\!]\}.$$

In Proposition 1, we provide sufficient conditions that guarantee that the ordinal policy is β-smooth, when the predictor, g_ω, belongs to a large class of functions, such as some neural networks. When the predictor is a linear function, the obtained policy is β-smooth as long as the feature mapping is bounded (Corollary 1).

2.2 Linear prediction

If the function g_ω is a linear function, in ω, then the obtained policy is exactly equal to the cumulative ordinal regression model, presented by Agresti [3]. Indeed, let us suppose $\Omega = \mathbb{R}^p$ and let us consider a feature mapping $\phi \colon \mathcal{S} \to \mathbb{R}^p$, then suppose the function $g_\omega(\cdot)$ is a linear function: $g_\omega(s) = \langle \phi(s), \omega \rangle_{\mathbb{R}^p}$. Then, with Equation 2, we obtain:

$$\mathbb{P}(A \leq j) = \sigma\left(\tau_j - \langle \phi(s), \omega \rangle_{\mathbb{R}^d}\right),$$

which is the definition of a logistic cumulative ordinal regression model [3].

2.3 Thresholded model

The presented ordinal policy does not need the predictor $g_\omega(\cdot)$ to be a linear function. Indeed, in its more general form, the presented ordinal in an instance of thresholded model [14]. To use it with policy gradient methods, the predictor g_ω should be differentiable in its parameter ω, and preferable somehow "smooth" in its parameter. For instance, it could be parametrized by any neural network, this would allow solving complex tasks using DRL techniques. The obtained policy is then the same as a CORAL or CORN neural networks [4, 20].

2.4 Sufficient conditions for β-smoothness on ordinal policies

We now present sufficient conditions to guarantee that an ordinal policy is β-smooth. This is important because theoretical results [1] then ensure that policy gradient methods can be applied to improve the policy.

Proposition 1. *Let $s \in \mathcal{S}$, if:*

- *The function $\omega \mapsto g_\omega(s)$ is L_Ω-Lipschitz*
- *The function $\omega \mapsto \nabla_\omega g_\omega(s)$ is C_Ω-Lipschitz function and bounded by M_Ω*

Then the function:

$$\begin{aligned} \Omega \times \Delta_{K-1}^{\varepsilon} &\to \mathbb{R} \\ (\omega, \tau) &\mapsto \log \pi_{(\tau,\omega)}(a, s) \end{aligned}$$

is β-smooth, with $\beta = \sqrt{2D_\varepsilon^2 + C_\varepsilon^2}$, where $C_\varepsilon = (1 + \exp(-\varepsilon))^{-2}$ and $D_\varepsilon = \sqrt{2}\max(C_\Omega + \sqrt{2}M_\Omega C_\varepsilon L_\Omega, M_\Omega)$.

We provide a proof of Property 1 as supplementary material

As a direct consequence of Property 1, we obtain sufficient conditions assuring that an ordinal policy with a linear predictor is β-smooth. We present these condition in Corollary 1.

Corollary 1. *If $g_\omega(.) = \langle \phi(.), \omega \rangle_{\mathbb{R}^p}$ and $\|\phi(s)\| \leq M_\phi$, then the ordinal policy π_θ is β-smooth with $\beta = \sqrt{2D_\varepsilon^2 + C_\varepsilon^2}$, where $C_\varepsilon = (1 + \exp(-\varepsilon))^{-2}$ and $D_\varepsilon = \sqrt{2}\max(\sqrt{2}M_\phi^2 C_\varepsilon, M_\phi)$.*

2.5 Implementation details

It is difficult to use policy optimization methods over the set Δ_K^{ε} because there are constraints that must be respected to remain over this parametric set: the thresholds must remain ordered and the difference between thresholds must be controlled, this guarantees stability (Proposition 1). We tackle this by using a reparametrization $\xi = (\xi_1, \xi_2, \ldots, \xi_{K-1})$ of $\tau = (\tau_1, \tau_2, \ldots, \tau_{K-1})$, with:

$$\xi_1 = \tau_1 \quad ; \quad \xi_{k+1} = \log(\tau_{k+1} - \tau_k - \varepsilon), \quad k \in [\![1, K-2]\!].$$

Thus, policy improvement can be done on (ω, ξ) which belong to the unconstrained space $\Omega \times \mathbb{R}^{K-1}$. Then the corresponding thresholds, $(\tau_k)_{k=1}^{K-1}$, can be computed using the inverse of this reparametrization.

Additionally, using log-probabilities instead of probabilities is numerically more stable. Let $a \in [\![2, K-1]\!]$ and $\eta_k = \tau_k - g_\omega(s)$, for ordinal policies we can use the exact expression of log-probabilities given by:

$$\begin{aligned} &\log \pi_\theta(1|s) = \log \sigma(\eta_1), \quad \log \pi(K|s) = \log \sigma(-\eta_{K-1}), \\ &\log \pi(a|s) = g_\omega(s) + \log(\exp(-\tau_{a-1}) - \exp(-\tau_a)) + \log \sigma(\eta_a) + \log \sigma(\eta_{a-1}), \end{aligned}$$

2.6 Multivariate ordinal actions

It is straightforward to extend univariate ordinal actions to multivariate ordinal actions, by factorizing across action dimensions. Indeed, let $d \in \mathbb{N}$, let us consider a d-dimensional continuous action space $\mathcal{A} = [\![1, K]\!]^d$. We define a predictor function $g_\omega \colon \mathcal{S} \to \mathbb{R}^d$ and consider a parameter $\tau \in (\Delta_K^\varepsilon)^d$. Then for a given state $s \in \mathcal{S}$, let us note:

$$g_\omega(s) = (g_\omega^{(1)}(s), \dots, g_\omega^{(d)}(s)) \quad ; \quad \tau = \left(\tau_k^{(j)}\right)_{k \in [\![1,K]\!], j \in [\![1,d]\!]}.$$

Let $(k^{(1)}, \dots k^{(d)}) \in \mathcal{A}$, we define the multivariate cumulative distribution function of a policy by the following relationship:

$$\mathbb{P}(A_1 \leq k^{(1)}, \dots, A \leq k^{(d)}) = \prod_{j=1}^{d} \sigma\left(\tau_{k^{(j)}}^{(j)} - g_\omega^{(j)}(s)\right),$$

Which induces a conditional probability distribution over $\mathcal{A}$. It is worth noting that actions are not drawn independently across dimensions because the distribution depends on the multivariate predictor g_ω.

3 Applications

3.1 A simple simulation setting

A key point to consider is that the presented ordinal policy is more restrictive than the usual softmax policy. As an alternative to an ordinal policy, the policy may be parametrized by a univariate softmax distribution, but then the prediction zones for the different classes may not be ordered. In this section, we study numerically if this has an impact on the rate of improvement or quality of the policies in a simple simulation setting. This simulation setting is similar to the scenario we expect to observe when facing the problem described in Section 1.2.

We simulate the ALS of one episode by sampling from a Gaussian process, which is then squished into the set $[0, 5]$ using a sigmoid function. Additionally, we simulate the user response using a fixed, but unknown ordinal model π_U. Let us suppose that at a given instant, for a given state measure s_t, the class a_t is proposed. We keep track of a "discomfort" score Z_t, which is updated:

$$Z_{t+1} = (1 - \pi_U(a_t|s_t))^{\gamma_{\text{reaction}}} + \gamma_{\text{discomfort}} Z_t.$$

Then, with probability $\sigma(Z_{t+1})$, the user reacts, if he does, a tint class is drawn accordingly to $\pi_U(.|s_t)$. The term $(1 - \pi_U(a_t|s_t))$ measures the discrepancy between the proposed class and user preferred class. The environment parameter $\gamma_{\text{reaction}} \in \mathbb{R}^+$ determines the "laziness" of the user: if it is low, the user often reacts, if it is high, the user only reacts if the proposed class is unlikely to be drawn by π_U. At last, the environment parameter $\gamma_{\text{discomfort}} \in [0, 1]$ determines

the memory of the user, if this parameter is zero, the user only reacts to the current proposed class.

Using this simulation setting, we compare the performances of ordinal policies against the performances of softmax policies using as updates: REINFORCE, NPG and TRPO. For every policy and every method, the hyperparameters are tuned, to provide a fair comparison among parametrizations.

We use $\gamma_{\text{reaction}} = 0.5$, $\gamma_{\text{discomfort}} = 1$ and a discount rate of 0.9. We simulate episodes of length sixty, at the end of each episode the policy is updated. For a given update strategy and policy, we simulate four hundred episodes. This process gives one learning trajectory, and we simulate ten learning trajectories per update strategy and policy combinations. We present the simulation results in Figure 1. Similarly to how using ordinal information provides better predic-

Fig. 1. Total episodic reward vs episode, for different policy gradient methods and policies. Average curves are calculated using mean of ten random seeds, calculated on learning curves after a rolling mean with a window of twenty episodes. Shaded areas show mean ± one standard deviation.

tions in a supervised setting [7], we find that using ordinal policies improves the performance of policy gradient methods, independently of the method that is used. This suggests that using an ordinal policy, when there is a notion of order among actions.

Indeed, convergence seems faster and towards a better policy for ordinal policies than for multinomial policies, and this parametrization seems more stable: the standard deviation is smaller when an ordinal policy is used (see Figure 1).

Among the six studied methods, updating an ordinal policy using TRPO yield the best results. Indeed, it provides a faster improvement rate than RE-

INFORCE while being slightly more stable than NPG, this is consistent with numerical experiments presented by Schulman *et al.* [18].

We should stress that the simulation setting may favor ordinal policies, as tints are chosen from an ordinal model. Nevertheless, the setting reflects what is expected in a real setting: users tend to prefer clear tints for low luminosity and darker ones for high luminosity, while the thresholds are unknown.

3.2 A real life study

A study was done to test adapting the parameters of the hybrid tint control, described in Section 1.2, using an ordinal policy updated with TRPO.

We used an electrochromic prototype for this study, which was equipped with an ALS and two buttons in the branches of the frame. We implemented the hybrid mode controlling the tint using only the ALS values; the tint was automatically controlled by default, but at any moment the user could manually change the tint by using the buttons. After a user interaction, the automatic control was deactivated for ten seconds, and afterward the tint was again automatically controlled.

Nine users were recruited for this study; these reported using sunglasses in all seasons of the year and did not work at EssilorLuxottica nor in any optic-related company. All the participants consented to participate in the study, and a safety assessment validated the use of this electrochromic prototype in a controlled environment.

A walking circuit was created for this study, allowing users to experience various real-life light situations: indoor/outdoor transitions, reading and far vision situations, etc. The same circuit was used for all participants, and participants walked the circuit one at a time. Participants were free to walk the circuit at their rhythm; on average, the circuit was completed after 10 minutes. Each user was asked to walk the circuit four times while wearing the electrochromic frames in the hybrid tint control. At the end of each circuit, the parameters controlling the hybrid mode were updated using TRPO. The same maximum Kullback-Leibler divergence was used for every participant after every circuit.

The objective of this study was to assess whether the number of user interactions decreases as the model parameters are updated.

For one of the nine users, the model parameters could not be updated; the TRPO algorithm proposed the old parameters after each episode. We discuss why this is so later in this section. We present the results obtained with the eight remaining participants in Figure 2. Two participants did not press the button at any moment of the study; after the four circuits were done, they reported that the proposed tint was well adapted for them. At first glance, the number of button presses seems to decrease as the model is updated, for the remaining participants.

To assess if there was a significant decrease in the number of user interactions, we analyzed the trend using a Poisson regression, modeling the link between the number of button pressings per episode Y *(count response)* in function of the episode number X *(numeric covariate)* with an additive interaction for the user

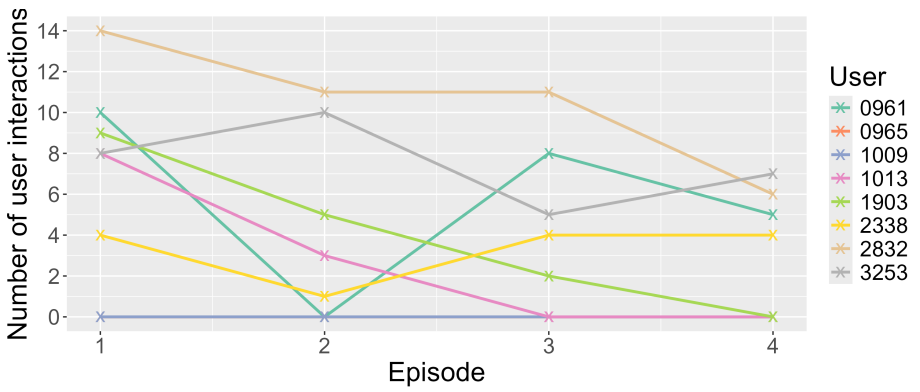

Fig. 2. Number of button pressings, per episode and user. Button pressings corresponding to a darker (resp. lighter) tint when current tint is C_3 (resp. C_0) were removed

j (categorical covariate) to account for the differences between users. Concretely, we consider the statistical model:

$$Y \sim \text{Poisson}(\lambda);\ \log(\lambda) = \alpha \cdot X + F_j,$$

where the slope $\alpha \in \mathbb{R}$ and user effects $\{F_j\}_{j=1}^{6}$ are model parameters. We present the observed and fitted number of button pressings as well as 95% confidence prediction intervals in Figure 3. The estimated slope α is statistically significantly negative (point estimation: -0.28 and 95% confidence interval: $[-0.43, -0.13]$). To address possible overfitting issues caused by the observed sparse dataset, we calculate a confidence interval for α using a leave-one-out procedure, the *jackknife* method [5], and obtain a point estimation of -0.28 and a 95% confidence interval of $[-0.49, -0.06]$. Since the slope α is statistically significantly negative, the number of button pressings diminishes as circuits go by.

These are encouraging results for this exploratory study, yet because no control group was used, we cannot claim causality; we cannot claim that users interact less often *because* the model parameters are updated. A confirmatory study with more users and a control group should be done to confirm this.

After a close inspection of the data of the participant for whom the model did not update, we noticed an unexpected behavior: instead of reacting to the proposed tints, this participant tried different tints and then chose the most adapted one. This behavior is not well described by the RL formulation of Section 1.2. For instance, when ALS was high, the tint C_3 was proposed, and then the user tried all the tints and then manually selected the tint C_3. Thus, proposing the tint C_3 was a good choice in reality. Instead, with our RL approach, proposing the tint C_3 would be judged as a bad action because it yields low rewards: ALS would be high, the tint C_3 would be proposed, and then the user would successively press the button six times. This is likely the reason for which TRPO could not update the model parameters.

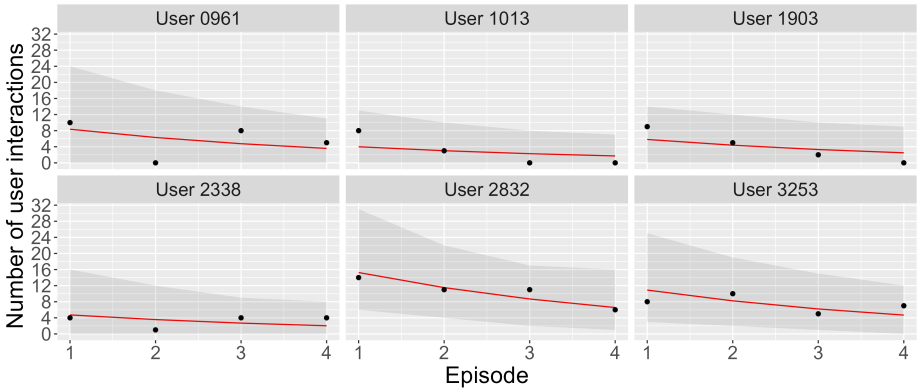

Fig. 3. Observed number of user interaction, fitted average (red line) and 95% prediction intervals, per user and episode.

A simple solution to align the RL formulation with this behavior is to adapt the reward signal. For example, instead of having a reward of -1 when there is a user interaction, the reward could be the opposite of the eventual difference between the user selected and automatically proposed tints.

3.3 Discretizing continuous state space

Ordinal actions may arise from the discretization of a continuous action space. Indeed, let us consider a bounded continuous action space $\mathcal{A} = [m, M]$. Let us consider the following discretization of the action space:

$$\mathcal{A}_K = \{a_k\}_{k=1}^{K} \quad ; \quad a_k = m + k\frac{M-m}{K}.$$

The set $\mathcal{A}_K$ is an ordinal set: there is an immediate order relationship between actions, Equation 1, and there is a finite number of actions. Hence, we may take actions on $\mathcal{A}_K$ using ordinal policies.

When multivariate continuous are considered, the same discretization may be applied dimension-wise and then use the multivariate ordinal extension presented in Section 2.6. We implement this approach, using an ordinal policy with 17 classes (per dimension) to solve different Mujoco [23] and other benchmark RL environments. To improve the policy we use PPO, as implemented for PPO for continuous actions by Huang *et al.* [10], with the same hyperparameters. We use the same neural network parametrizing PPO for continuous actions implemented by Huang *et al.* [10], which uses a two layer MLP to parametrize $g_\omega(\cdot)$ and uses a learnable standard deviation per action dimension, independent of the state, as suggested by Huang *et al.* [9]. We run experiments in the environments of the Table 1, using a learning rate of $3 \cdot 10^{-4}$ and a discount factor of $\gamma = 0.99$ for both continuous action PPO and ordinal PPO, in all the environments.We obtain the results presented in Figure 4

Environment	State dimensions	Action Dimensions
Ant-v4	105	8
BipedalWalker-v3	24	4
HalfCheetah-v4	17	6
Hopper-v4	11	3
Humanoid-v4	348	17
InvertedDoublePendulum-v4	9	1
Pusher-v4	23	7
Walker2d-v4	17	6

Table 1. Continuous action environments where ordinal actions were used by discretization

We find that using ordinal policies for classic RL problems with continuous actions yields similar performances than using a continuous policy. This is coherent with results from the literature [19, 22].

4 Conclusion — Discussion

Satisfying every user, using one predefined model with fixed parameters, is challenging since each user's preferences are unique and may vary and evolve. Thanks to today's highly configurable wearable devices, we may tailor a model to answer each individual's needs. In this article, we present a method to do this in an online manner using the RL paradigm. This opens a door for a wide range of real applications, but in a real-world setting, it is important to do so robustly. For instance, when the system pilots an ordinal setting, using an adapted policy provides this robustness: no matter how the user interacts with the wearable, the policy will always be an ordinal model, thus the notion of order between levels is assured.

Furthermore, the studied simulation setting suggests that considering the notion of order, when there is one, is beneficial. And the proposed method can be directly applied in a real-world setting (Section 3.2) and a deep learning approach may also be used (Section 3.3).

There are two main axes for future work. For the industrial application, it is of great importance to converge fast to a "good" model. Indeed, a method that would require thousands of episodes to improve would be pointless for any real-life application: the user may simply stop wearing the device before there are enough episodes. Thus, a solution would be to use simple models, such as the one used in Section 3.2, or leverage off-policy methods, which are known to be more sample-efficient than on-policy methods. Secondly, ALS collected during a time window could better explain user context and thus allow for better policies than the ones that use only the instantaneous ALS. This temporal covariate could be used with an adapted ordinal model, such as the one proposed by Jacques and Samardžić [11].

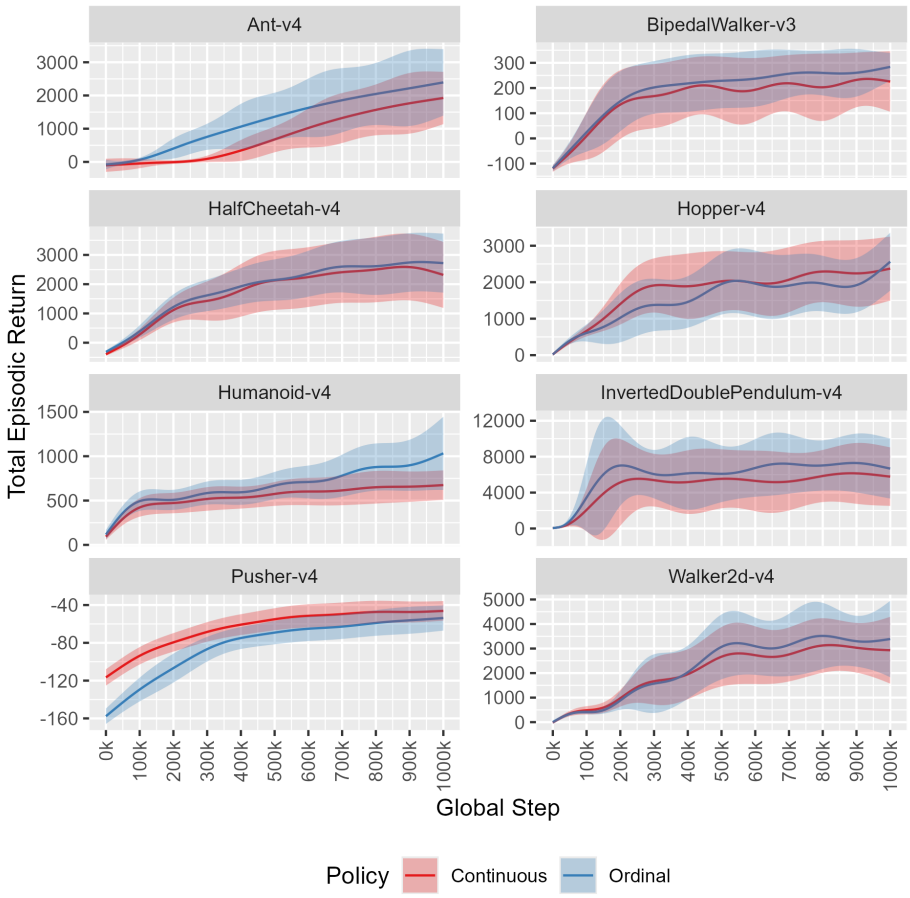

Fig. 4. Learning curves for different policies across different environments. Mean and 95% confidence prediction interval, fitted using a location-scale Gaussian GAM on data generated using five different random seeds per environment.

Acknowledgments. We thank Walid M'Zough, Khalil BEN GHORBEL, Carole NADOLNY, Armel JIMENEZ, Alexandre GOURRAUD and Jean SAHLER for their collaboration and support for the study presented in this article.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Bibliography

[1] Agarwal, A., Kakade, S.M., Lee, J., Mahajan, G.: On the theory of policy gradient methods: Optimality, approximation, and distribution shift. J. Mach. Learn. Res. **22**, 98:1–98:76 (2019)

[2] Agarwal, A., Kakade, S.M., Lee, J.D., Mahajan, G.: Optimality and approximation with policy gradient methods in markov decision processes. In: Abernethy, J., Agarwal, S. (eds.) Proceedings of Thirty Third Conference on Learning Theory. Proceedings of Machine Learning Research, vol. 125, pp. 64–66. PMLR (09–12 Jul 2020)

[3] Agresti, A.: Analysis of ordinal categorical data, vol. 656. John Wiley & Sons (2010)

[4] Cao, W., Mirjalili, V., Raschka, S.: Rank consistent ordinal regression for neural networks with application to age estimation. Pattern Recognition Letters **140**, 325–331 (2020)

[5] Efron, B., Tibshirani, R.J.: An introduction to the bootstrap. Chapman and Hall/CRC (1994)

[6] Feng, J., Shi, Y., Qu, G., Low, S.H., Anandkumar, A., Wierman, A.: Stability constrained reinforcement learning for decentralized real-time voltage control. IEEE Transactions on Control of Network Systems **11**(3), 1370–1381 (2024). https://doi.org/10.1109/TCNS.2023.3338240

[7] Gutiérrez, P.A., Pérez-Ortiz, M., Sánchez-Monedero, J., Fernández-Navarro, F., Hervás-Martínez, C.: Ordinal regression methods: Survey and experimental study. IEEE Transactions on Knowledge and Data Engineering **28**(1), 127–146 (2016)

[8] Henriksen, A., Haugen Mikalsen, M., Woldaregay, A.Z., Muzny, M., Hartvigsen, G., Hopstock, L.A., Grimsgaard, S.: Using fitness trackers and smartwatches to measure physical activity in research: analysis of consumer wrist-worn wearables. Journal of medical Internet research **20**(3), e110 (2018)

[9] Huang, S., Dossa, R.F.J., Raffin, A., Kanervisto, A., Wang, W.: The 37 implementation details of proximal policy optimization. In: ICLR Blog Track (2022)

[10] Huang, S., Dossa, R.F.J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., Araújo, J.G.: Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. Journal of Machine Learning Research **23**(274), 1–18 (2022), http://jmlr.org/papers/v23/21-1342.html

[11] Jacques, J., Samardžić, S.: Analyzing cycling sensors data through ordinal logistic regression with functional covariates. Journal of the Royal Statistical Society: Series C Applied Statistics (2022), https://hal.archives-ouvertes.fr/hal-03107427

[12] Khan, M., Mohamed, S., Marlin, B., Murphy, K.: A stick-breaking likelihood for categorical data analysis with latent gaussian models. In: Lawrence,

N.D., Girolami, M. (eds.) Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 22, pp. 610–618. PMLR, La Palma, Canary Islands (21–23 Apr 2012)

[13] Kim, J., Campbell, A.S., de Ávila, B.E.F., Wang, J.: Wearable biosensors for healthcare monitoring. Nature biotechnology **37**(4), 389–406 (2019)

[14] Li, L., Lin, H.t.: Ordinal regression by extended binary classification. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems. vol. 19. MIT Press (2006)

[15] McCullagh, P.: Regression models for ordinal data. Journal of the Royal Statistical Society. Series B (Methodological) **42**(2), 109–142 (1980), http://www.jstor.org/stable/2984952

[16] Mukhopadhyay, S.C.: Wearable sensors for human activity monitoring: A review. IEEE Sensors Journal **15**, 1321–1330 (2015)

[17] Sabry, F., Eltaras, T., Labda, W., Alzoubi, K., Malluhi, Q.: Machine learning for healthcare wearable devices: The big picture. Journal of Healthcare Engineering **2022**(1), 4653923 (2022). https://doi.org/https://doi.org/10.1155/2022/4653923, https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/4653923

[18] Schulman, J., Levine, S., Moritz, P., Jordan, M.I., Abbeel, P.: Trust region policy optimization. CoRR **abs/1502.05477** (2015)

[19] Seyde, T., Gilitschenski, I., Schwarting, W., Stellato, B., Riedmiller, M., Wulfmeier, M., Rus, D.: Is bang-bang control all you need? solving continuous control with bernoulli policies. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 27209–27221. Curran Associates, Inc. (2021)

[20] Shi, X., Cao, W., Raschka, S.: Deep neural networks for rank-consistent ordinal regression based on conditional probabilities (2021)

[21] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. A Bradford Book, Cambridge, MA, USA (2018)

[22] Tang, Y., Agrawal, S.: Discretizing continuous action space for on-policy optimization. CoRR **abs/1901.10500** (2019), http://arxiv.org/abs/1901.10500

[23] Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5026–5033. IEEE (2012). https://doi.org/10.1109/IROS.2012.6386109

[24] Tranchina, D., Gordon, J., Shapley, R.: Retinal light adaptation—evidence for a feedback mechanism. Nature **310**(5975), 314–316 (1984)

[25] Wehenkel, A., Louppe, G.: Unconstrained monotonic neural networks. Advances in neural information processing systems **32** (2019)

[26] Weinberger, S., Cugliari, J., Le Cain, A.: Ordinal regression for preference learning in wearables using sensor data. Expert Systems with Applications **281**, 127616 (2025)

[27] Wu, Z., Khan, N.M., Gao, L., Guan, L.: Deep reinforcement learning with parameterized action space for object detection. In: 2018 IEEE International Symposium on Multimedia (ISM). pp. 101–104 (2018). https://doi.org/10.1109/ISM.2018.00025

[28] Yee, T.W., Wild, C.J.: Vector Generalized Additive Models. Journal of the Royal Statistical Society: Series B (Methodological) **58**(3), 481–493 (12 1996). https://doi.org/10.1111/j.2517-6161.1996.tb02095.x