Stop Regressing: Training Value Functions via Classification for Scalable Deep RL

Jesse Farebrother [*] [1] [2] **Jordi Orbay** [1] **Quan Vuong** [1] **Adrien Ali Taïga** [1] **Yevgen Chebotar** [1] **Ted Xiao** [1]
Alex Irpan [1] **Sergey Levine** [1] **Pablo Samuel Castro** [1] [3] **Aleksandra Faust** [1] **Aviral Kumar** [1] **Rishabh Agarwal** [*] [1] [3]

Abstract

Value functions are an essential component in deep reinforcement learning (RL), that are typically trained via mean squared error regression to match bootstrapped target values. However, scaling value-based RL methods to large networks has proven challenging. This difficulty is in stark contrast to supervised learning: by leveraging a cross-entropy classification loss, supervised methods have scaled reliably to massive networks. Observing this discrepancy, in this paper, we investigate whether the scalability of deep RL can also be improved simply by using classification in place of regression for training value functions. We show that training value functions with categorical cross-entropy significantly enhances performance and scalability across various domains, including single-task RL on Atari 2600 games, multi-task RL on Atari with large-scale ResNets, robotic manipulation with Q-transformers, playing Chess without search, and a language-agent Wordle task with high-capacity Transformers, achieving *state-of-the-art results* on these domains. Through careful analysis, we show that categorical cross-entropy mitigates issues inherent to value-based RL, such as noisy targets and non-stationarity. We argue that shifting to categorical cross-entropy for training value functions can substantially improve the scalability of deep RL at little-to-no cost.

1. Introduction

A clear pattern emerges in deep learning breakthroughs – from AlexNet (Krizhevsky et al., 2012) to Transformers (Vaswani et al., 2017) – classification problems seem to be particularly amenable to effective training with large neural networks. Even when regression appears natural, reframing the problem as one of classification often improves performance (Torgo & Gama, 1996; Rothe et al., 2018; Rogez et al., 2019). This involves converting real-valued targets into categorical labels and minimizing categorical cross-entropy rather than the mean-squared error. Several hypotheses have been proposed to explain the superiority of this approach, including stable gradients (Imani & White, 2018; Imani et al., 2024), better representations (Zhang et al., 2023), implicit bias (Stewart et al., 2023), and handling imbalanced data (Pintea et al., 2023) – suggesting potential utility beyond supervised regression.

Unlike trends in supervised learning, value-based reinforcement learning (RL) methods primarily rely on regression. For example, deep RL methods such as deep Q-learning (Mnih et al., 2015) and actor-critic (Mnih et al., 2016) use a regression loss, such as mean-squared error, to train a value function from continuous scalar targets. While these value-based deep RL methods, powered by regression losses, have led to high-profile results (Silver et al., 2017), it has been challenging to scale them up to large networks, such as high-capacity transformers. This lack of scalability has been attributed to several issues (Kumar et al., 2021; 2022; Agarwal et al., 2021; Lyle et al., 2022; Le Lan et al., 2023), but ***what if simply reframing the regression problem as classification can enable the same level of scalability achieved in supervised learning?***

In this paper, we extensively study the efficacy of various methods for deriving classification labels for training a value-function with a categorical cross-entropy loss. Our findings reveal that training value-functions with cross-entropy substantially improves the performance, robustness, and scalability of deep RL methods compared to traditional regression-based approaches. The most notable method (HL-Gauss; Imani & White, 2018) leads to consist performance improvements in single-task RL on Atari; $\mathbf{1.8 - 2.1\times}$ performance in multi-task setups on Atari (Kumar et al., 2023; Ali Taïga et al., 2023); $\mathbf{40}$**%** better performance in the language-agent task of Wordle (Snell

[*] Equal contribution [1] Google DeepMind [2] Mila, McGill University [3] Mila, Université de Montréal. Correspondence to: Jesse Farebrother <jfarebro@cs.mcgill.ca>, Aviral Kumar <aviralkumar@google.com>, Rishabh Agarwal <rishabhagarwal@google.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

et al., 2023); **70%** improvement for playing chess without search (Ruoss et al., 2024); and **67%** better performance on large-scale robotic manipulation with transformers (Chebotar et al., 2023). The consistent trend across diverse domains, network architectures, and algorithms highlights the substantial benefits of treating regression as classification in deep RL, underscoring its potential as a pivotal component as we move towards scaling up value-based RL.

With **strong empirical results to support the use of cross-entropy as a "drop-in" replacement for the mean squared error (MSE) regression loss in deep RL**, we also attempt to understand the source of these empirical gains. Based on careful diagnostic experiments, we show that the categorical cross-entropy loss offers a number of benefits over mean-squared regression. Our analysis suggests that the categorical cross-entropy loss mitigates several issues inherent to deep RL, including robustness to noisy targets and allowing the network to better use its capacity to fit non-stationary targets. These findings not only help explain the strong empirical advantages of categorical cross-entropy in deep RL but also provide insight into developing more effective learning algorithms for the field.

2. Preliminaries and Background

Regression as classification. We take a probabilistic view on regression where given input $x \in \mathbb{R}^d$ we seek to model the target as a conditional distribution $Y \,|\, x \sim \mathcal{N}(\mu = \hat{y}(x;\theta), \sigma^2)$ for some fixed variance σ^2 and predictor function $\hat{y} : \mathbb{R}^d \times \mathbb{R}^k \to \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^k$. The maximum likelihood estimator for data $\{x_i, y_i\}_{i=1}^N$ is characterized by the **mean-squared error (MSE)** objective,

$$\min_\theta \sum_{i=1}^N \left(\hat{y}(x_i;\theta) - y_i\right)^2 ,$$

with the optimal predictor being $\hat{y}(x;\theta^*) = \mathbb{E}\left[Y \,|\, x\right]$.

Instead of directly learning the mean of the conditional distribution, an alternate approach is to learn a distribution over the target value and recover the prediction $\hat{y}$ as a statistic of this distribution. To this end, we will construct the target distribution $Y \,|\, x$ with probability density function $p(y \,|\, x)$ such that the scalar target y is the mean of this distribution, $y = \mathbb{E}_p\left[Y \,|\, x\right]$. We can now frame the regression problem as learning a parameterized distribution $\hat{p}(y \,|\, x;\theta)$ that minimizes the KL divergence to the target $p(y \,|\, x)$,

$$\min_\theta \sum_{i=1}^N \int_{\mathcal{Y}} p(y \,|\, x_i) \log\left(\hat{p}(y \,|\, x_i;\theta)\right) dy \tag{2.1}$$

which is the cross-entropy objective. Finally, our prediction can be recovered as $\hat{y}(x;\theta) = \mathbb{E}_{\hat{p}}\left[\,Y \,|\, x;\theta\,\right]$.

Given this new problem formulation, in order to transform the distribution learning problem into a tractable loss we restrict $\hat{p}$ to the set of categorical distributions supported on $[v_{\text{min}}, v_{\text{max}}]$ with m evenly spaced locations or "classes", $v_{\text{min}} \le z_1 < \cdots < z_m \le v_{\text{max}}$ defined as,

$$\mathcal{Z} = \left\{ \sum_{i=1}^m p_i \, \delta_{z_i} \;:\; p_i \ge 0, \sum_{i=1}^m p_i = 1 \right\}, \tag{2.2}$$

where p_i is the probability associated with location z_i and δ_{z_i} is the Dirac delta function at location z_i. The final hurdle is to construct the target distribution $Y \,|\, x$ and its associated projection onto the set of categorical distributions $\mathcal{Z}$. We defer this discussion to §3 where we discuss various methods for performing these steps in the context of RL.

Reinforcement Learning (RL). We consider the reinforcement learning (RL) problem where an agent interacts with an environment by taking an action $A_t \in \mathcal{A}$ in the current state $S_t \in \mathcal{S}$ and subsequently prescribed a reward $R_{t+1} \in \mathbb{R}$ before transitioning to the next state $S_{t+1} \in \mathcal{S}$ according to the environment transition probabilities. The return numerically describes the quality of a sequence of actions as the cumulative discounted sum of rewards $G_t = \sum_{k=0}^\infty \gamma^k R_{t+k+1}$ where $\gamma \in [0, 1)$ is the discount factor. The agent's goal is to learn the policy $\pi : \mathcal{S} \to \mathscr{P}(\mathcal{A})$ that maximizes the expected return. The action-value function allows us to query the expected return from taking action a in state s and following policy π thereafter: $q_\pi(s, a) = \mathbb{E}_\pi\left[G_t \,|\, S_t = s,\, A_t = a\right]$.

Deep Q-Networks (DQN; Mnih et al., 2015) proposes to learn the approximately optimal state-action value function $Q(s, a;\theta) \approx q_{\pi^*}(s, a)$ with a neural network parameterized by θ. Specifically, DQN minimizes the mean-squared temporal difference (TD) error from transitions (S, A, R, S') sampled from dataset $\mathcal{D}$,

$$\boxed{\text{TD}_{\text{MSE}}(\theta) = \mathbb{E}_{\mathcal{D}}\left[\left((\widehat{\mathcal{T}}Q)(S, A;\tilde{\theta}) - Q(S, A;\theta)\right)^2\right]} \tag{2.3}$$

where $\tilde{\theta}$ is a slow moving copy of the parameters θ that parameterize the "target network" and

$$(\widehat{\mathcal{T}}Q)(s, a;\tilde{\theta}) = R + \gamma \max_{a'} Q(S', a';\tilde{\theta}) \;\big|\; S = s, A = a\,,$$

is the sample version of the Bellman optimality operator which defines our scalar regression target. Most deep RL algorithms leveraging value functions follow this basic recipe, notably regressing to predictions from a target network.

We also explore the offline RL setting where an agent is trained using a fixed dataset of environment interactions (Agarwal et al., 2020; Levine et al., 2020). One widely-used offline RL method is conservative Q-learning (CQL; Kumar et al., 2020) that jointly optimizes TD_{MSE} with the following behavior regularization loss scaled by $\alpha \in \mathbb{R}$,

$$\alpha\, \mathbb{E}_{\mathcal{D}}\left[\log\Big(\sum_{a'} \exp(Q(S', a';\theta))\Big) - Q(S, A;\theta)\right]. \tag{2.4}$$

3. Value-Based RL with Classification

In this section, we describe our approach to cast the regression problem appearing in TD-learning as a classification problem. Concretely, instead of minimizing the squared distance between the scalar Q-value and its TD target (2.3) we will instead minimize the distance between categorical distributions representing these quantities. To employ this approach, we will first define the categorical representation for the action-value function $Q(s, a)$.

Categorical Representation. We choose to represent Q as the expected value of a categorical distribution $Z \in \mathcal{Z}$. This distribution is parameterized by probabilities $\hat{p}_i(s, a; \theta)$ for each location or "class" z_i which are derived from the logits $l_i(s, a; \theta)$ through the softmax function:

$$Q(s, a; \theta) = \mathbb{E}\left[\, Z(s, a; \theta) \,\right], \; Z(s, a; \theta) = \sum_{i=1}^{m} \hat{p}_i(s, a; \theta) \cdot \delta_{z_i},$$

$$\hat{p}_i(s, a; \theta) = \frac{\exp\left(l_i(s, a; \theta)\right)}{\sum_{j=1}^{m} \exp\left(l_j(s, a; \theta)\right)}.$$

To employ the cross-entropy loss (2.1) for TD learning, we must define a target categorical distribution supported on the same locations $z_i, \ldots, z_m$ such that $\sum_{i=1}^{m} p_i(S, A; \tilde{\theta})\, z_i \approx (\widehat{\mathcal{T}}Q)(S, A; \tilde{\theta})$ with p_i being the target probabilities. This enables the direct computation of the cross-entropy loss as,

$$\boxed{\mathrm{TD}_{\mathrm{CE}}(\theta) = \mathbb{E}_{\mathcal{D}}\left[\sum_{i=1}^{m} p_i(S, A; \tilde{\theta}) \log \hat{p}_i(S, A; \theta)\right]} \tag{3.1}$$

In the subsequent sections, we explore two strategies for obtaining the target probabilities $p_i(S, A; \tilde{\theta})$.

3.1. Categorical Distributions from Scalars

The first set of methods we outline will project the scalar target $(\widehat{\mathcal{T}}Q)(S, A; \tilde{\theta})$ onto the categorical distribution supported on $\{z_i\}_{i=1}^{m}$. A prevalent but naïve approach for the projection step involves discretizing the scalar into one of m bins where z_i represents the center of the bin. The resulting one-hot distribution is "lossy" and induces errors in the Q-function. These errors would compound as more Bellman backups are performed, resulting in more biased estimates, and likely worse performance. To combat this, we first consider the "two-hot" approach (Schrittwieser et al., 2020) that represents a scalar target *exactly* via a unique categorical distribution that puts non-zero densities on two locations that the target lies between (see Figure 1; Left).

A Two-Hot Categorical Distribution. Let z_i and z_{i+1} be the locations which lower and upper-bound the TD target $y = (\widehat{\mathcal{T}}Q)(S, A; \tilde{\theta})$, i.e., $z_i \leq y \leq z_{i+1}$. Then, the probability, p_i and p_{i+1}, put on these locations is:

$$p_i(S, A; \tilde{\theta}) = \frac{y - z_i}{z_{i+1} - z_i}, \; p_{i+1}(S, A; \tilde{\theta}) = \frac{z_{i+1} - y}{z_{i+1} - z_i}. \tag{3.2}$$

For all other locations, the probability prescribed by the categorical distribution is zero. In principle, this Two-Hot transformation provides a uniquely identifiable and a non-lossy representation of the scalar TD target to a categorical distribution. However, Two-Hot does not fully harness the ordinal structure of discrete regression. Specifically, the classes are not independent and instead have a natural ordering, where each class intrinsically relates to its neighbors.

The class of Histogram Losses introduced by Imani & White (2018) seeks to exploit the ordinal structure of the regression task by distributing probability mass to neighboring bins – akin to label smoothing in supervised classification (Szegedy et al., 2016). This is done by transforming a noisy version of the target into a categorical distribution, allowing probability mass to span multiple bins near the target (See Figure 1; Center), rather than being limited to two locations.

Histograms as Categorical Distributions. Formally, define the random variable $Y \,|\, S, A$ with probability density $f_{Y|S,A}$ and cumulative distribution function $F_{Y|S,A}$ whose expectation is $(\widehat{\mathcal{T}}Q)(S, A; \tilde{\theta})$. We can project the distribution $Y \,|\, S, A$ onto the histogram with bins of width $\varsigma = (v_{\text{max}} - v_{\text{min}})/m$ centered at z_i by integrating over the interval $[z_i - \varsigma/2, z_i + \varsigma/2]$ to obtain the probabilities,

$$\begin{aligned} p_i(S, A; \tilde{\theta}) &= \int_{z_i - \varsigma/2}^{z_i + \varsigma/2} f_{Y|S,A}(y|S, A)dy \\ &= F_{Y|S,A}(z_i + \varsigma/2|S, A) - F_{Y|S,A}(z_i - \varsigma/2|S, A). \end{aligned} \tag{3.3}$$

We now have a choice for the distribution $Y \,|\, S, A$. We follow the suggestion of Imani & White (2018) in using the Gaussian distribution $Y \,|\, S, A \sim \mathcal{N}(\mu = (\widehat{\mathcal{T}}Q)(S, A; \tilde{\theta}), \sigma^2)$ where the variance σ^2 controls the amount of label smoothing applied to the resulting categorical distribution. We refer to this method as HL-Gauss.

How should we tune σ in practice? HL-Gauss requires tuning the standard deviation σ, in addition to the bin width ς and distribution range $[v_{min}, v_{max}]$. 99.7% of the samples obtained by sampling from a standard Normal distribution should lie within three standard deviations of the mean with high confidence, which corresponds to approximately $6 \cdot \sigma/\varsigma$ bins. Thus, a more interpretable hyper-parameter that we recommend tuning is σ/ς: setting it to $K/6$ distributes most of the probability mass to $\lceil K \rceil + 1$ neighbouring locations for a mean value centered at one of the bins. Unless specified otherwise, we set $\sigma/\varsigma = 0.75$ for our experiments, which distributes mass to approximately 6 locations.

3.2. Modelling the Categorical Return Distribution

In the previous section, we constructed a target distribution from the usual scalar regression target representing the expected return. Another approach is to model the distribution over future returns directly using our categorical model Z,

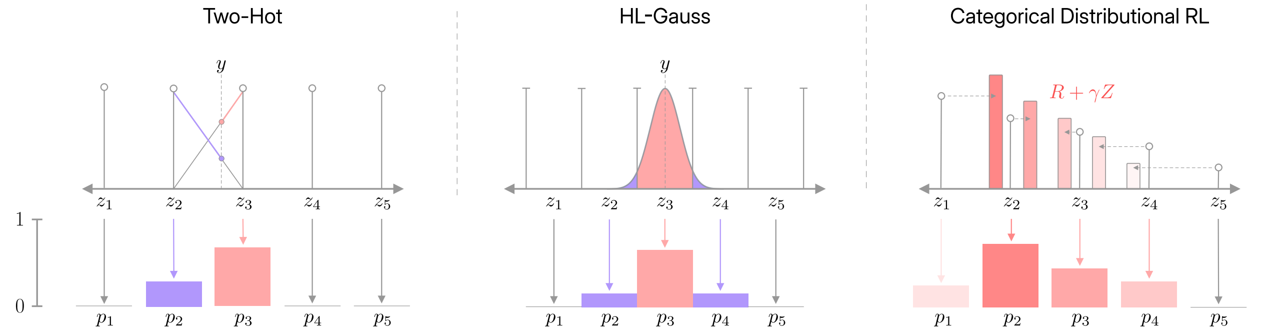

Figure 1. **Visualizing categorical distributions in cross-entropy based TD learning**. Two-Hot (left, §3.1) puts probability mass on exactly two locations. HL-Gauss (middle, §3.1) distributes the probability mass to neighbouring locations (akin to smoothing the target value). CDRL (right, §3.2) models the categorical return distribution, distributing probability mass proportionally to neighboring locations.

as done in distributional RL (Bellemare et al., 2023). Notably, C51 (Bellemare et al., 2017), an early distributional RL method, uses the categorical representation and minimizes the cross-entropy between the predicted distribution Z and the distributional TD target. We also investigate C51 as an alternative to Two-Hot and HL-Gauss for constructing the target distribution for the cross-entropy objective (3.1).

Categorical Distributional RL. The first step to modelling the categorical return distribution is to define the analogous stochastic distributional Bellman operator on Z,

$$(\widehat{\mathcal{T}}Z)(s,a;\tilde{\theta}) \overset{D}{=} \sum_{i=1}^{m} \hat{p}_i(S',A';\tilde{\theta}) \cdot \delta_{R+\gamma z_i} \;\Big|\; S=s,\, A=a\,,$$

where $A' = \arg\max_{a'} Q(S',a';\tilde{\theta})$. As we can see, the stochastic distributional Bellman operator has the effect of shifting and scaling the locations z_i necessitating the categorical projection, first introduced by Bellemare et al. (2017). At a high level, this projection distributes probabilities proportionally to the immediate neighboring locations $z_{j-1} \le R_{t+1} + \gamma z_i \le z_j$ (See Figure 1; Right). To help us identify these neighboring locations we define $\lfloor x \rfloor = \arg\max\{z_i : z_i \le x\}$ and $\lceil x \rceil = \arg\min\{z_i : z_i \ge x\}$. Now the probabilities for location z_i can be written as,

$$p_i(S,A;\tilde{\theta}) = \sum_{j=1}^{m} \hat{p}_j(S',A';\tilde{\theta}) \cdot \xi_j(R+\gamma z_i) \qquad (3.4)$$

$$\xi_j(x) = \frac{x - z_j}{z_{j+1} - z_j} \mathbb{1}\{\lfloor x \rfloor = z_j\} + \frac{z_{j+1} - x}{z_{j+1} - z_j} \mathbb{1}\{\lceil x \rceil = z_j\}\,.$$

For a complete exposition of the categorical projection, see Bellemare et al. (2023, Chapter 5).

4. Evaluating Classification Losses in RL

4.1. Single-Task RL on Atari Games

The goal of this section is to evaluate the efficacy of the various target distributions discussed in Section 3 combined with the categorical cross-entropy loss (3.1) in improving performance and scalability of value-based deep RL on a variety of problems. This includes several single-task and multi-task RL problems on Atari 2600 games as well as domains beyond Atari including language agents, chess, and robotic manipulation. These tasks consist of both online and offline RL problems. For each task, we instantiate our cross-entropy losses in conjunction with a strong value-based RL approach previously evaluated on that task. Full experimental methodologies including hyperparameters for each domain we consider can be found in Appendix C.

We first evaluate the efficacy of HL-Gauss, Two-Hot, and C51 (Bellemare et al., 2017), on the Arcade Learning Environment (Bellemare et al., 2013). For our regression baseline we train DQN (Mnih et al., 2015) on the mean-squared error TD objective which has been shown to outperform other regression based losses (Obando-Ceron & Castro, 2021). Each method is trained with the Adam optimizer (Kingma & Ba, 2015), which has been shown to reduce the performance discrepancy between regression-based methods and distributional RL approaches (Agarwal et al., 2021).

Evaluation. Following the recommendations by Agarwal et al. (2021), we report the interquartile mean (IQM) normalized scores with 95% stratified bootstrap confidence intervals (CIs), aggregated across games with multiple seeds each. We report human-normalized aggregated scores across 60 Atari games for online RL. For offline RL, we report behavior-policy normalized scores aggregated across 17 games, following the protocol in Kumar et al. (2021).

Online RL results. Following Mnih et al. (2015), we train DQN for 200M frames with the aforementioned losses. We report aggregated human-normalized IQM performance and optimality gap across 60 Atari games in Figure 2. Observe that HL-Gauss substantially outperforms the Two-Hot and MSE losses. Interestingly, HL-Gauss also improves upon categorical distributional RL (C51), despite not modelling the return distribution. This finding suggests that the loss (categorical cross-entropy) is perhaps the more crucial factor for C51, as compared to modelling the return distribution.

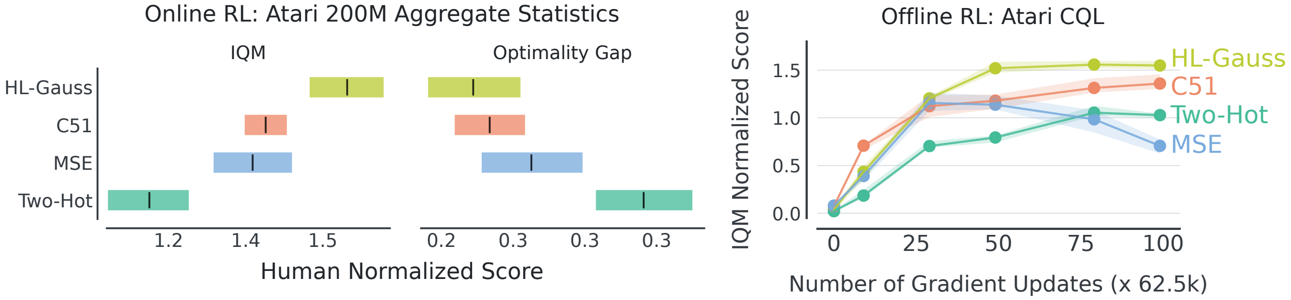

Figure 2. **Regression vs Classification for (Left) Online and (Right) Offline RL.** HL-Gauss and CDRL outperform MSE, with HL-Gauss performing best. Moreover, Two-Hot loss underperforms MSE but remains stable with prolonged training in offline RL, akin to other cross-entropy losses. We report aggregated scores with 95% CIs across 60 games for online RL and 17 games for offline RL.

Offline RL results. The strong performance of HL-Gauss with online DQN, which involves learning from self-collected interactions, raises the question of its effectiveness when learning from offline datasets. To explore this, we train agents with different losses on the 10% Atari DQN replay dataset (Agarwal et al., 2020) using CQL (2.4) for 6.25M gradient steps. As shown in Figure 2, HL-Gauss and C51 consistently outperform MSE, while Two-Hot shows improved stability over MSE but underperforms other classification methods. Notably, HL-Gauss again surpasses C51 in this setting. Additionally, as found by Kumar et al. (2021), using the mean squared regression loss leads to performance degradation with prolonged training. However, cross-entropy losses (both HL-Gauss and C51) do not show such degradation and generally, remain stable.

4.2. Scaling Value-Based RL to Large Networks

In supervised classification, especially language modeling (Kaplan et al., 2020), increasing the network's parameter count usually improves performance. However, such scaling remains elusive for value-based deep RL, where *naive* parameter scaling can hurt performance (Ali Taïga et al., 2023; Kumar et al., 2023; Obando-Ceron et al., 2024b;a). Noticing this discrepancy we now explore if our classification methods for learning value-functions in deep RL can achieve similar performance gains when scaling parameters.

Multi-task Online RL. Following Ali Taïga et al. (2023), we train a multi-task policy capable of playing Atari game variants with different environment dynamics and rewards (Farebrother et al., 2018). We evaluate two Atari games: 63 variants of ASTEROIDS and 29 variants of SPACE INVADERS. We employ the distributed actor-critic method IMPALA (Espeholt et al., 2018), and compare the standard MSE critic loss with the cross-entropy based HL-Gauss loss. Our experiments investigate the scaling properties of these losses from the Impala-CNN ($\leq$ 2M parameters) to larger ResNets (He et al., 2016) up to ResNet-101 (44M parameters). We evaluate multi-task performance after training for 15 billion frames, repeating each experiment with 5 seeds.

Results for ASTEROIDS are presented in Figure 3, with additional results on SPACE INVADERS presented in Figure D.4. We observe that in both environments HL-Gauss consistently outperforms MSE. Notably, HL-Gauss scales better, especially on ASTEROIDS where it even slightly improves performance with larger networks beyond ResNet-18, while MSE performance significantly degrades.

Multi-game Offline RL. We adapt the setup from Kumar et al. (2023), by replacing the distributional RL based C51 loss with the non-distributional HL-Gauss loss. Specifically, we train a single generalist policy to play 40 different Atari games simultaneously, when learning from a "near-optimal" training dataset, composed of replay buffers obtained from online RL agents trained independently on each game. This multi-game RL setup was initially proposed by Lee et al. (2022). All other design choices, such as feature normalization and network size, remain unchanged.

As shown in Figure 3, HL-Gauss scales even better than the C51 results from Kumar et al. (2023), resulting in an improvement of about 45% over the best prior multi-game result available with ResNet-101 (80M parameters) as measured by the IQM human normalized score (Agarwal et al., 2021). Furthermore, while the performance of MSE regression losses typically plateaus upon increasing model capacity beyond ResNet-34, HL-Gauss is able to leverage this capacity to improve performance, indicating the efficacy of classification-based cross-entropy losses. Additionally, when normalizing against scores obtained by a DQN agent, we show in Figure D.5 that in addition to performance, the rate of improvement as the model scale increases tends to also be larger for the HL-Gauss loss compared to C51.

4.3. Value-Based RL with Transformers

Next, we evaluate the applicability of the HL-Gauss cross-entropy loss beyond Atari. To do so, we consider several tasks that utilize high-capacity Transformers, namely, a language-agent task of playing Wordle, playing Chess without inference-time search, and robotic manipulation.

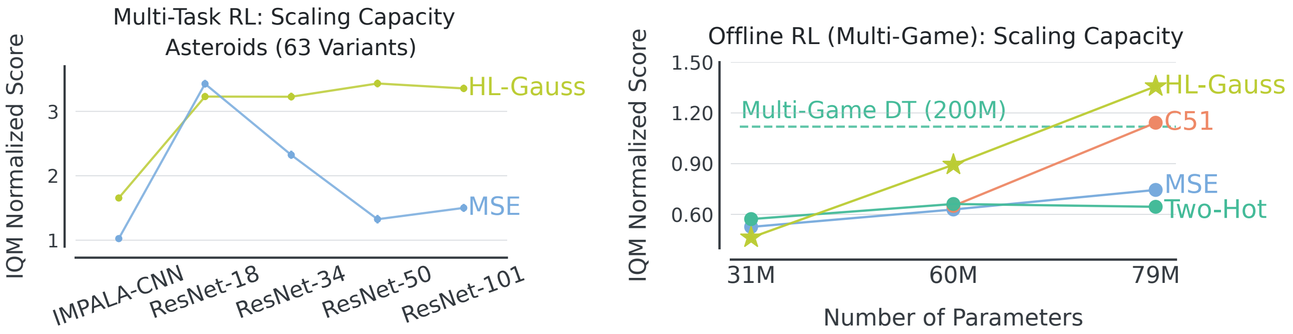

Figure 3. **Scaling Curves on Multi-task RL. (Left)** Results for IMPALA with ResNets on ASTEROIDS. Lacking human scores we report the IMPALA normalized IQM. HL-Gauss outperforms MSE and scales better with larger networks. **(Right)** IQM human normalized score for ResNet-$\{34, 50, 101\}$, playing 40 Atari games simultaneously (Kumar et al., 2023). HL-Gauss significantly imporves scaling, outperforming categorical distributional RL (C51), regression (MSE), and the multi-game Decision Transformer (Lee et al., 2022).

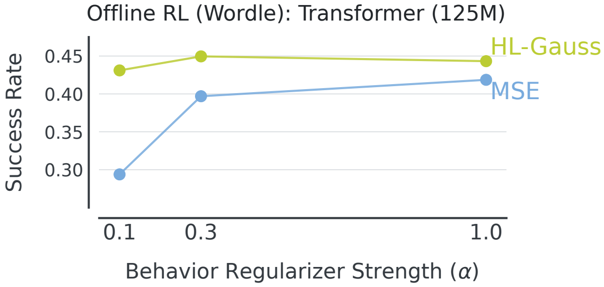

Figure 4. **Language Agent.** Comparing HL-Gauss with MSE for a Transformer trained with offline RL on Wordle games (Snell et al., 2023). HL-Gauss achieves higher success rates in guessing the word in one turn, across varying behavior regularization strengths.

Wordle. To evaluate whether classification losses improve value-based RL performance on language agent benchmarks, we compare HL-Gauss with MSE on the task of playing the game of Wordle. Wordle is a word guessing game in which the agent gets 6 attempts to guess a word. Each turn the agent receives environment feedback about whether guessed letters are in the true word. The dynamics of this task are non-deterministic. More generally, the task follows a turn-based structure, reminiscent of dialogue tasks in natural language processing. This experiment is situated in the offline RL setting, where we utilize the dataset of suboptimal game-plays provided by Snell et al. (2023). Our goal is to train a GPT-like, decoder-only Transformer, with 125M parameters, representing the Q-network.

On this task, we train the language-based transformer for 20K gradient steps with an offline RL approach combining Q-learning updates from DQN with a CQL-style behavior regularizer (2.4), which corresponds to standard next-token prediction loss (in this particular problem). As shown in Figure 4, HL-Gauss outperforms MSE, for multiple coefficients controlling the strength of CQL regularization.

Grandmaster-level Chess. Transformers have demonstrated their effectiveness as general-purpose algorithm approximators, effectively amortizing expensive inference-time computation through distillation (Ruoss et al., 2024; Lehnert et al., 2024). In this context, we explore the potential benefits of using HL-Gauss to convert scalar action-values into classification targets for distilling a value-function. Using the setup of Ruoss et al. (2024), we evaluate HL-Gauss for distilling the action-value function of Stockfish 16 — the strongest available Chess engine that uses a combination of complex heuristics and explicit search — into a causal transformer. The distillation dataset comprises 10 million chess games annotated by the Stockfish engine, yielding 15 billion data points. Appendix C.3 provides additional details on the dataset curation in Ruoss et al. (2024).

We train 3 transformer models of varying capacity (9M, 137M, and 270M parameters) on this dataset, using either HL-Gauss or 1-Hot classification targets. We omit MSE as Ruoss et al. (2024) demonstrate that 1-Hot targets outperform MSE on this task. The effectiveness of each model is evaluated based on its ability to solve 10,000 chess puzzles from Lichess, with success measured by the accuracy of the generated action sequences compared to known solutions. Both the setup and results are presented in Figure 5. While the one-hot target with the 270M Transformer from Ruoss et al. (2024) outperformed an AlphaZero baseline without search, HL-Gauss closes the performance gap with the substantially stronger AlphaZero with 400 MCTS simulations (Schrittwieser et al., 2020).

Generalist Robotic Manipulation. Finally, we evaluate whether cross-entropy losses can improve performance on a set of large-scale vision-based robotic manipulation control tasks from Chebotar et al. (2023). These tasks present a simulated 7-DoF mobile manipulator, placed in front of a countertop surface. The goal is to control this manipulator to successfully grasp and lift 17 different kitchen objects in

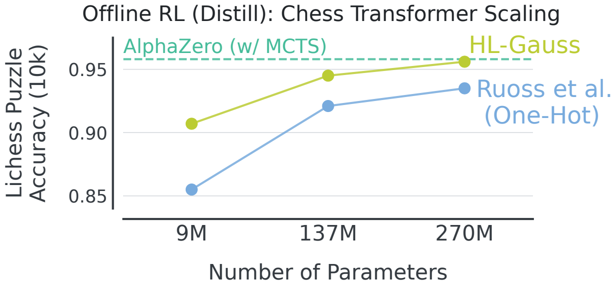

Figure 5. **Grandmaster-level Chess without Search.** Following the setup from Ruoss et al. (2024), where they train Transformer models to play chess via supervised learning on Stockfish 16 Q-values and then follow greedy policy for evaluation. HL-Gauss outperforms one-hot targets used by Ruoss et al. (2024) and nearly matches the performance of AlphaZero with tree search.

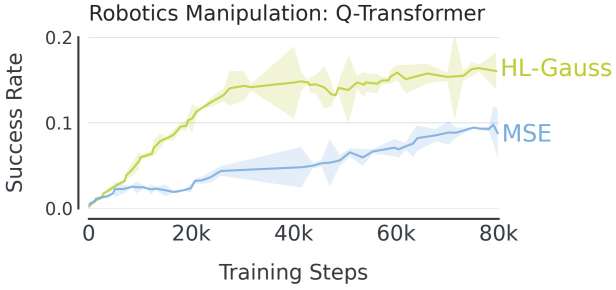

Figure 6. **Generalist robotic manipulation with offline data: HL-Gauss vs MSE on simulated vision-based manipulation.** Robotic manipulation using a 7 degree of freedom mobile manipulator robot from Chebotar et al. (2023). In the plots, error bars show 95% CIs. Note that utilizing a HL-Gauss enables significantly faster learning to a better point.

the presence of distractor objects, clutter, and randomized initial poses. We generate a dataset of $500,000$ (successful and failed) episodes starting from a small amount of human-teleoperated demonstrations ($40,000$ episodes) by replaying expert demonstrations with added sampled action noise, reminiscent of failed autonomously-collected rollouts obtained during deployment or evaluations of a behavioral cloning policy trained on the human demonstration data.

We train a Q-Transformer model with 60M parameters, following the recipe in Chebotar et al. (2023), but replace the MSE regression loss with the HL-Gauss classification loss. As shown in Figure 6, HL-Gauss results in 67% higher peak performance over the regression baseline, while being much more sample-efficient, addressing a key limitation of the prior regression-based approach.

5. Why Does Classification Benefit RL?

Our experiments demonstrate that classification losses can significantly improve the performance and scalability of value-based deep RL. In this section, we perform controlled experiments to understand why classification benefits value-based RL. Specifically, we attempt to understand how the categorical cross-entropy loss can address several challenges specific to value-based RL including representation learning, stability, and robustness. We will also perform ablation experiments to uncover the reasons behind the superiority of HL-Gauss over other categorical targets.

5.1. What are the Mechanisms of Classification Losses?

Classification losses presented in this paper differ from traditional regression losses used in value-based RL in two ways: **(1)** parameterizing the output of the value-network to be a categorical distribution in place of a scalar, and **(2)** strategies for converting scalar targets into a categorical target. We will now understand the relative contribution of these steps towards the performance of cross-entropy losses.

Are Categorical Representations More Performant? As discussed in §3.1, we parameterize the Q-network to output logits that are converted to probabilities using the "softmax" operator. Softmax leads to bounded Q-values and output gradients, which can possibly improve RL training stability (Hansen et al., 2024). To investigate whether our Q-value parameterization alone results in improved performance without needing a cross-entropy loss, we train Q-functions with the same parameterization as (3.1) but with MSE. We find no gains from using softmax with MSE in both online (Figure 9) and offline RL (Figure D.6). This shows that cross-entropy is key to the performance improvements.

Why Are Certain Cross-Entropy Losses Better? Our results show that HL-Gauss outperforms Two-Hot, even though both use a cross-entropy loss. We hypothesize that HL-Gauss benefits from: 1) reduced overfitting by spreading probability mass to neighboring locations; and 2) generalization across a specific range of target values, exploiting ordinal structure in the regression problem. Notably, the first hypothesis aligns with how label smoothing addresses overfitting in classification problems (Szegedy et al., 2016).

We test these hypotheses in the online RL setting across a subset of 13 Atari games. To do so, we fix the value range $[v_{\min}, v_{\max}]$ while varying the number of bins in $\{21, 51, 101, 201\}$ and the ratio of standard deviation σ to bin width ς in $\{0.25, 0.5, 0.75, 1.0, 2.0\}$. Figure 11 shows that HL-Gauss outperform Two-Hot across a wide range of σ values, suggesting reduced overfitting due to the spread of probability mass to neighboring locations. Interestingly, we notice that the second hypothesis is also at play, as the optimal value of σ seems to be independent of the number of bins, indicating that HL-Gauss generalizes best across a specific range of target values leveraging the ordinal nature of the regression problem. Thus, we conclude the gains from HL-Gauss are not solely due to overfitting, as is believed to be the case for label smoothing in supervised learning.

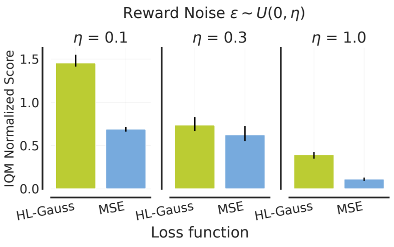

Figure 7. Comparing HL-Gauss and MSE when trained using noisy rewards in offline RL on Atari, across 17 games.

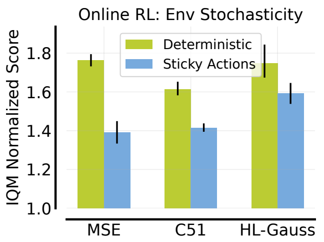

Figure 8. Impact of stochastic dynamics on cross-entropy and regression losses in online RL across 60 games.

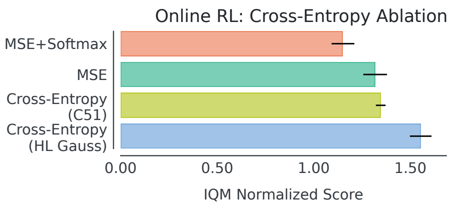

Figure 9. Categorical representation of Q-values (§3.1) does not benefit MSE loss, implying that the cross-entropy loss is critical.

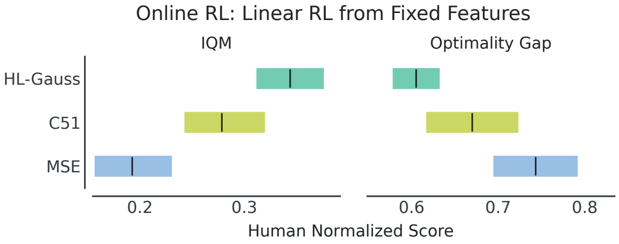

Figure 10. **Evaluating representations using linear probing on Atari.** Optimality gap refers to the distance from human-level performance and lower is better. In both plots, the representations learned by HL-Gauss are more conducive to policy optimization.

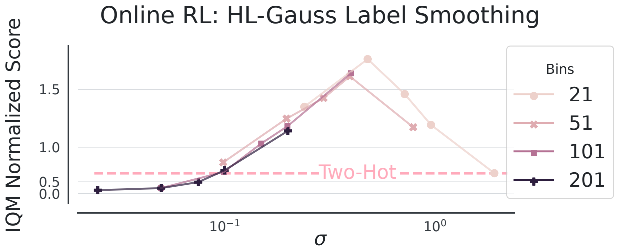

Figure 11. **Sweeping the ratio σ/ς for different number of bins in Online RL on Atari.**. HL-Gauss benefits from label smoothing with the optimal amount of label smoothing as prescribed by σ being independent of bin width ς.

5.2. What Challenges Does Classification Tackle in RL?

Having seen that the performance gains of cross-entropy losses stem from both the use of a categorical representation of values and distributed targets, we now attempt to understand which challenges in value-based RL are addressed, or at least partially alleviated, by cross-entropy losses.

Is Classification More Robust to Noisy Targets? Classification is less prone to overfitting to noisy targets than regression, as it focuses on categorical rather than numerical relationships between the input and target. To explore this further, we investigate whether classification can better handle noise induced by stochasticity in RL.

(a) Noisy Rewards. To test robustness of classification to stochasticity in rewards, we consider an offline RL setup where we add random noise ε_t, sampled uniformly from $(0, \eta)$, to each dataset reward r_t. We vary the noise scale $\eta \in \{0.1, 0.3, 1.0\}$ and compare the performance of cross-entropy based HL-Gauss with the MSE loss. As shown in Figure 7, the performance of HL-Gauss degrades more gracefully than MSE as the noise scale increases.

(b) Stochasticity in Dynamics. Following Machado et al. (2018), our Atari experiments use sticky actions — with 25% probability, the environment will execute the previous action again, instead of the agent's intended action — resulting in non-deterministic dynamics. Here, we turn off sticky actions to compare different losses on deterministic Atari (60 games). As shown in Figure 8, while cross-entropy based HL-Gauss outperforms MSE with stochastic dynamics, they perform comparably under deterministic dynamics while outperforming categorical distributional RL (C51).

Overall, the benefits of cross-entropy losses can be partly attributed to less overfitting to noisy targets, an issue inherent to RL environments with stochastic dynamics or rewards. Such stochasticity issues may also arise as a result of dynamics mis-specification or action delays in real-world embodied RL problems, implying that a cross-entropy loss is a superior choice in those problems.

Does Classification Learn More Expressive Representations? It is well known that just using the mean-squared regression error alone does not produce useful representations in value-based RL, often resulting in low capacity representations (Kumar et al., 2021) that are incapable of fitting target values observed during subsequent training (Lyle et al., 2022). Predicting a categorical distribution rather than a scalar target can lead to better representations (Zhang et al., 2023), that retain the representational power to model value functions of arbitrary policies that might be encountered over the course of value learning (Dabney et al., 2021). Lyle et al. (2019) showed that gains from C51 can be partially attributed to improved representations but it remains unknown whether they stem from backing up distributions of returns or the use of cross-entropy loss.

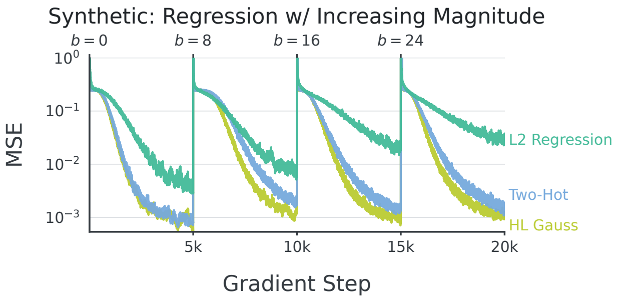

Figure 12. **Synthetic magnitude experiment.** Non-stationarity is simulated by fitting high-frequency targets on an increasing sequences of biases. Classification is less likely to lose plasticity.

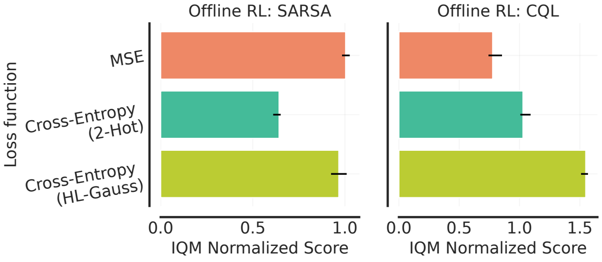

Figure 13. **Offline QL vs SARSA on Atari.** HL-Gauss gains over MSE dissapear with SARSA, suggesting that classification benefits from addressing policy non-stationarity.

To investigate this question, following the protocol in Farebrother et al. (2023), we study whether a learned representation, corresponding to the penultimate feature vector, obtained from value-functions trained online on Atari for 200M frames, still retains the necessary information to re-learn a policy from scratch. To do so, we train a Q-function with a single linear layer on top of this frozen representation, akin to how self-supervised representations are evaluated in vision (He et al., 2020). As shown in Figure 10, cross-entropy losses result in better performance with linear probing. This indicates that their learned representations are indeed better in terms of supporting the value-improvement path of a policy trained from scratch (Dabney et al., 2021).

Is Classification More Robust to Non-Stationarity? Non-stationarity is inherent to value-based RL as the target computation involves a constantly evolving argmax policy and value function. Bellemare et al. (2017) hypothesized that classification might mitigate difficulty of learning from a non-stationary policy, but did not empirically validate it. Here, we investigate whether classification can indeed handle target non-stationarity better than regression.

We first consider a synthetic regression task on CIFAR10 from Lyle et al. (2024), where the regression target maps an image x_i through a randomly initialized convolutional network f_{θ^-} producing high-frequency targets $y_i = \sin(10^5 \cdot f_{\theta^-}(x_i)) + b$, where b is bias controlling for the magnitude of the targets. In TD learning, prediction targets are non-stationary and often increase in magnitude as the policy improves. We simulate this setting by fitting a network with different losses on the increasing sequence of biases $b \in \{0, 8, 16, 24, 32\}$. See details in Appendix C.5. As shown in Figure 12, classification losses retain higher plasticity under non-stationary targets compared to regression.

In the context of RL, we can control for policy non-stationarity by performing offline SARSA following the protocol in Kumar et al. (2022). That is, when computing the Bellman target $(\widehat{\mathcal{T}}Q)(S, A)$, SARSA uses an in-distribution sample of the observed action at the next timestep (S', A'). In contrast, Q-learning uses the action that maximizes the Q-value at S' which introduces additional non-stationarity. Figure 13 shows that most of the benefit from HL-Gauss compared to the MSE vanishes in the offline SARSA setting, adding evidence that some of the benefits from classification stem from dealing with non-stationarity of the target policy.

In summary, we find that the use of cross-entropy loss itself is central to obtain good performance in value-based RL, and while these methods do not address any specific challenge, they enable value-based RL methods to deal better with non-stationarity, induce highly-expressive representations, and provide robustness against noisy target values.

6. Conclusion

In this paper, we showed that framing regression as classification and minimizing categorical cross-entropy instead of the mean squared error significantly improves the performance and scalability of value-based RL methods across a wide variety of tasks and neural network architectures. We analyzed the source of these improvements and found that they stem from the ability of the cross-entropy loss in enabling more expressive representations and better handling of noise and non-stationarity in value-based RL. While cross-entropy alone does not completely solve these issues, our results highlight the substantial benefits of this change.

We believe that strong results with the use categorical cross-entropy has implications for future algorithm design in deep RL, both in theory and practice. Practically, value-based RL approaches have been harder to scale and tune when the value function is represented by a transformer architecture and our results hint that classification might provide for a smooth approach to translate innovation in value-based RL to transformers. Theoretically, analyzing the optimization dynamics of cross-entropy might help devise improved losses or target distribution representations. Finally, while we did explore a number of settings, further work is required to evaluate the efficacy of classification losses in other RL problems such as pre-training, fine-tuning, or continual RL.

Acknowledgements

We would like to thank Will Dabney for providing feedback on an early version of this paper. We'd also like to thank Clare Lyle, Mark Rowland, Marc Bellemare, Max Schwarzer, Pierluca D'oro, Nate Rahn, Harley Wiltzer, Wesley Chung, and Dale Schuurmans, for informative discussions. We'd also like to acknowledge Anian Ruoss, Grégoire Delétang, and Tim Genewein for their help with the Chess training infrastructure. This research was supported by the TPU resources at Google DeepMind, and the authors are grateful to Doina Precup and Joelle Baral for their support.

We would also like to thank the Python community whose contributions made this work possible. In particular, this work made extensive use of Numpy (Harris et al., 2020), SciPy (Virtanen et al., 2020), Jax (Bradbury et al., 2018), Flax (Heek et al., 2023), Optax (DeepMind et al., 2020), matplotlib (Hunter, 2007), and Seaborn (Waskom, 2021).

Author Contributions

JF led the project, implemented histogram-based methods, ran all the single-task online RL experiments on Atari, Q-distillation on Chess, jointly proposed and ran most analysis experiments, and contributed significantly to writing.

JO and AAT set up and ran the multi-task RL experiments and helped with writing. QV ran the robotic manipulation experiments and YC helped with the initial set-up. TX helped with paper writing and AI was involved in discussions. SL advised on the robotics and Wordle experiments and provided feedback. PSC helped set up the SoftMoE experiments and hosted Jesse at GDM. PSC and AF sponsored the project and took part in discussions.

AK advised the project, proposed offline analysis for non-stationarity and representation learning, contributed significantly to writing, revising, and the narrative, and set up the robotics and multi-game scaling experiments. RA proposed the research direction, advised the project, led the paper writing, ran offline RL and Wordle experiments, and helped set up the multi-task scaling and non-Atari experiments.

Impact Statement

This paper presents work whose goal is to advance the field of Reinforcement learning and Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Achab, M., Alami, R., Djilali, Y. A. D., Fedyanin, K., and Moulines, E. One-step distributional reinforcement learning. *Transactions on Machine Learning Research (TMLR)*, 2023.

Agarwal, R., Schuurmans, D., and Norouzi, M. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2020.

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A., and Bellemare, M. G. Deep reinforcement learning at the edge of the statistical precipice. *Neural Information Processing Systems (NeurIPS)*, 2021.

Ali Taïga, A., Agarwal, R., Farebrother, J., Courville, A., and Bellemare, M. G. Investigating multi-task pretraining and generalization in reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Ayoub, A., Wang, K., Liu, V., Robertson, S., McInerney, J., Liang, D., Kallus, N., and Szepesvári, C. Switching the loss reduces the cost in batch reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2024.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research (JAIR)*, 47:253–279, 2013.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2017.

Bellemare, M. G., Dabney, W., Dadashi, R., Ali Taïga, A., Castro, P. S., Le Roux, N., Schuurmans, D., Lattimore, T., and Lyle, C. A geometric perspective on optimal representations for reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Bellemare, M. G., Dabney, W., and Rowland, M. *Distributional reinforcement learning*. MIT Press, 2023.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Carvalho, W., Saraiva, A., Filos, A., Lampinen, A. K., Matthey, L., Lewis, R. L., Lee, H., Singh, S., Rezende, D. J., and Zoran, D. Combining behaviors with the successor features keyboard. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Castro, P. S., Moitra, S., Gelada, C., Kumar, S., and Bellemare, M. G. Dopamine: A Research Framework for Deep Reinforcement Learning. *CoRR*, abs/1812.06110, 2018. URL `https://github.com/google/dopamine`.

Chebotar, Y., Vuong, Q., Hausman, K., Xia, F., Lu, Y., Irpan, A., Kumar, A., Yu, T., Herzog, A., Pertsch, K., et al. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *Conference on Robot Learning (CoRL)*, 2023.

Dabney, W., Barreto, A., Rowland, M., Dadashi, R., Quan, J., Bellemare, M. G., and Silver, D. The value-improvement path: Towards better representations for reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2021.

DeepMind, Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Dedieu, A., Fantacci, C., Godwin, J., Jones, C., Hemsley, R., Hennigan, T., Hessel, M., Hou, S., Kapturowski, S., Keck, T., Kemaev, I., King, M., Kunesch, M., Martens, L., Merzic, H., Mikulik, V., Norman, T., Papamakarios, G., Quan, J., Ring, R., Ruiz, F., Sanchez, A., Sartran, L., Schneider, R., Sezener, E., Spencer, S., Srinivasan, S., Stanojević, M., Stokowiec, W., Wang, L., Zhou, G., and Viola, F. The DeepMind JAX Ecosystem, 2020. URL `http://github.com/google-deepmind`.

D'Oro, P., Schwarzer, M., Nikishin, E., Bacon, P.-L., Bellemare, M. G., and Courville, A. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *International Conference on Learning Representations (ICLR)*, 2023.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., and Kavukcuoglu, K. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning (ICML)*, 2018.

Farebrother, J., Machado, M. C., and Bowling, M. Generalization and regularization in DQN. *CoRR*, abs/1810.00123, 2018.

Farebrother, J., Greaves, J., Agarwal, R., Le Lan, C., Goroshin, R., Samuel Castro, P., and Bellemare, M. G. Proto-value networks: Scaling representation learning with auxiliary tasks. In *International Conference on Learning Representations (ICLR)*, 2023.

Gordon, G. J. *Approximate Solutions to Markov Decision Processes*. PhD thesis, Carnegie Mellon University, 1999.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. P. Mastering diverse domains through world models. *CoRR*, abs/2301.04104, 2023.

Hansen, N., Su, H., and Wang, X. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Harris, C. R., K. Jarrod, M., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for JAX, 2023. URL `http://github.com/google/flax`.

Hessel, M., Danihelka, I., Viola, F., Guez, A., Schmitt, S., Sifre, L., Weber, T., Silver, D., and van Hasselt, H. Muesli: Combining improvements in policy optimization. In *International Conference on Machine Learning (ICML)*, 2021.

Ho, D., Rao, K., Xu, Z., Jang, E., Khansari, M., and Bai, Y. Retinagan: An object-aware approach to sim-to-real transfer. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

Imani, E. and White, M. Improving regression performance with distributional losses. In *International Conference on Machine Learning (ICML)*, 2018.

Imani, E., Luedemann, K., Scholnick-Hughes, S., Elelimy, E., and White, M. Investigating the histogram loss in regression. *CoRR*, abs/2402.13425, 2024.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020.

Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., and Bry, A. End-to-end learning of geometry and context for deep stereo regression. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Khakhar, A. and Buckman, J. Neural regression for scale-varying targets. *CoRR*, abs/2211.07447, 2023.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems (NeurIPS)*, 2012.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Neural Information Processing Systems (NeurIPS)*, 2020.

Kumar, A., Agarwal, R., Ghosh, D., and Levine, S. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Kumar, A., Agarwal, R., Ma, T., Courville, A., Tucker, G., and Levine, S. Dr3: Value-based deep reinforcement learning requires explicit regularization. In *International Conference on Learning Representations (ICLR)*, 2022.

Kumar, A., Agarwal, R., Geng, X., Tucker, G., and Levine, S. Offline Q-Learning on Diverse Multi-Task Data Both Scales and Generalizes. In *International Conference on Learning Representations (ICLR)*, 2023.

Le Lan, C., Tu, S., Oberman, A., Agarwal, R., and Bellemare, M. G. On the generalization of representations in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

Le Lan, C., Tu, S., Rowland, M., Harutyunyan, A., Agarwal, R., Bellemare, M. G., and Dabney, W. Bootstrapped representations in reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2023.

Lee, K.-H., Nachum, O., Yang, M. S., Lee, L., Freeman, D., Guadarrama, S., Fischer, I., Xu, W., Jang, E., Michalewski, H., and Mordatch, I. Multi-game decision transformers. In *Neural Information Processing Systems (NeurIPS)*, 2022.

Lehnert, L., Sukhbaatar, S., Mcvay, P., Rabbat, M., and Tian, Y. Beyond a*: Better planning with transformers via search dynamics bootstrapping. *CoRR*, abs/2402.14083, 2024.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *CoRR*, abs/2005.01643, 2020.

Lyle, C., Bellemare, M. G., and Castro, P. S. A comparative analysis of expected and distributional reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2019.

Lyle, C., Rowland, M., Ostrovski, G., and Dabney, W. On the effect of auxiliary tasks on representation dynamics. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Lyle, C., Rowland, M., and Dabney, W. Understanding and preventing capacity loss in reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Lyle, C., Zheng, Z., Khetarpal, K., van Hasselt, H., Pascanu, R., Martens, J., and Dabney, W. Disentangling the causes of plasticity loss in neural networks. *CoRR*, abs/2402.18762, 2024.

Machado, M. C., Bellemare, M. G., Talvitie, E., Veness, J., Hausknecht, M., and Bowling, M. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research (JAIR)*, 61:523–562, 2018.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2016.

Nikishin, E., Schwarzer, M., D'Oro, P., Bacon, P.-L., and Courville, A. The Primacy Bias in Deep Reinforcement Learning. In *ICML*, 2022.

Obando-Ceron, J., Courville, A., and Castro, P. S. In value-based deep reinforcement learning, a pruned network is a good network. In *International Conference on Machine Learning (ICML)*, 2024a.

Obando-Ceron, J., Sokar, G., Willi, T., Lyle, C., Farebrother, J., Foerster, J., Dziugaite, G. K., Precup, D., and Castro, P. S. Mixtures of experts unlock parameter scaling for deep rl. In *International Conference on Machine Learning (ICML)*, 2024b.

Obando-Ceron, J. S. and Castro, P. S. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *International Conference on Machine Learning (ICML)*, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Pintea, S. L., Lin, Y., Dijkstra, J., and van Gemert, J. C. A step towards understanding why classification helps regression. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 19972–19981, 2023.

Puigcerver, J., Ruiz, C. R., Mustafa, B., and Houlsby, N. From sparse to soft mixtures of experts. In *International Conference on Learning Representations (ICLR)*, 2024.

Rogez, G., Weinzaepfel, P., and Schmid, C. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 42(5):1146–1161, 2019.

Rothe, R., Timofte, R., and Van Gool, L. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, 126(2-4):144–157, 2018.

Rowland, M., Tang, Y., Lyle, C., Munos, R., Bellemare, M. G., and Dabney, W. The statistical benefits of quantile temporal-difference learning for value estimation. In *International Conference on Machine Learning (ICML)*, 2023.

Ruoss, A., Delétang, G., Medapati, S., Grau-Moya, J., Wenliang, L. K., Catt, E., Reid, J., and Genewein, T. Grandmaster-level chess without search. *CoRR*, abs/2402.04494, 2024.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T. P., and Silver, D. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Snell, C. V., Kostrikov, I., Su, Y., Yang, S., and Levine, S. Offline RL for natural language generation with implicit language q learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Sokar, G., Agarwal, R., Castro, P. S., and Evci, U. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2023.

Springenberg, J. T., Abdolmaleki, A., Zhang, J., Groth, O., Bloesch, M., Lampe, T., Brakel, P., Bechtle, S., Kapturowski, S., Hafner, R., et al. Offline actor-critic reinforcement learning scales to large models. *CoRR*, abs/2402.05546, 2024.

Stewart, L., Bach, F., Berthet, Q., and Vert, J.-P. Regression as classification: Influence of task formulation on neural network features. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Torgo, L. and Gama, J. Regression by classification. In *Brazilian Symposium on Artificial Intelligence*, pp. 51–60. Springer, 1996.

Van Den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International Conference on Machine Learning (ICML)*, 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Neural Information Processing Systems (NeurIPS)*, 2017.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

Wang, K., Zhou, K., Wu, R., Kallus, N., and Sun, W. The benefits of being distributional: Small-loss bounds for reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Waskom, M. L. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

Weiss, S. M. and Indurkhya, N. Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research (JAIR)*, 3:383–403, 1995.

Zhang, S., Yang, L., Mi, M. B., Zheng, X., and Yao, A. Improving deep regression with ordinal entropy. In *International Conference on Learning Representations (ICLR)*, 2023.

Appendices

A. Related Work

Prior works in tabular regression (Weiss & Indurkhya, 1995; Torgo & Gama, 1996; Khakhar & Buckman, 2023) and computer vision (Van Den Oord et al., 2016; Kendall et al., 2017; Rothe et al., 2018; Rogez et al., 2019) have replaced regression with classification to improve performance. Most notably, Imani & White (2018) proposed the HL-Gauss cross-entropy loss for regression and show its efficacy on small-scale supervised regression tasks, outside of RL. Within the body of work on RL, Ayoub et al. (2024) provides theoretical insights showing that replacing MSE with a cross-entropy based loss results in lower sample complexity when performing fitted-value iteration (Gordon, 1999). Our work complements these prior works by illustrating for the first time that a classification objective trained with cross-entropy, particularly HL-Gauss, can enable effectively scaling for value-based RL on a variety of domains, including Atari, robotic manipulation, chess, and Wordle.

Several state-of-the-art methods in RL have used the Two-Hot cross-entropy loss without any analysis, either as an "ad-hoc" trick (Schrittwieser et al., 2020), citing benefits for sparse rewards (Hafner et al., 2023), or simply relying on folk wisdom (Hessel et al., 2021; Carvalho et al., 2023; Hansen et al., 2024). However, in our experiments, Two-Hot performs worse than other cross-entropy losses and MSE. We believe this is because Two-Hot does not effectively distribute probability to neighboring classes, unlike C51 and HL-Gauss (see §5.1 for an empirical investigation).

Closely related is the line of work on categorical distributional RL. Notably, Achab et al. (2023) offer an analysis of categorical one-step distributional RL, which corresponds precisely to the Two-Hot algorithm discussed herein with the similarity of these two approaches not being previously recognized. Additionally, the work of Bellemare et al. (2017) pioneered the C51 algorithm, and while their primary focus *was not* on framing RL as classification, our findings suggest that the specific loss function employed may play a more significant role in the algorithm's success than modeling the return distribution itself. Several methods find that categorical distributional RL losses are important for scaling offline value-based RL (Kumar et al., 2023; Springenberg et al., 2024), but these works do not attempt to isolate which components of this paradigm are crucial for attaining positive scaling trends. We also note that these findings do not contradict recent theoretical work (Wang et al., 2023; Rowland et al., 2023) which argues that distributional RL brings statistical benefits over standard RL orthogonal to use of a cross entropy objective or the categorical representation.

Prior works have characterized the representations learned by TD-learning (Bellemare et al., 2019; Lyle et al., 2021; Le Lan et al., 2022; 2023; Kumar et al., 2021; 2022) but these prior works focus entirely on MSE losses with little to no work analyzing representations learned by cross-entropy based losses in RL. Our linear probing experiments in §5.2 try to fill this void, demonstrating that value-functions trained with cross-entropy losses learn better representations than regression. This finding is especially important since Imani & White (2018) did not find any representational benefits of HL-Gauss over MSE on supervised regression, indicating that the use of cross-entropy might have substantial benefits for TD-based learning methods in particular.

B. Reference Implementations

Both of the implementations listed below use the error function to evaluate the Gaussian CDF. This can be numerically unstable outside of the support hence we clip the values between the valid range. Both Two-Hot and Categorical Distributional RL performs a similar type of clipping but do allow for querying values outside of the support.

Listing 1 An implementation of HL-Gauss (Imani & White, 2018) in Jax (Bradbury et al., 2018).

```python
import jax
import jax.scipy.special
import jax.numpy as jnp


def hl_gauss_transform(
    min_value: float,
    max_value: float,
    num_bins: int,
    sigma: float,
):
    support = jnp.linspace(min_value, max_value, num_bins + 1, dtype=jnp.float32)

    def transform_to_probs(target: jax.Array) -> jax.Array:
        target = jnp.clip(target, min_value, max_value)
        cdf_evals = jax.scipy.special.erf((support - target) / (jnp.sqrt(2) * sigma))
        z = cdf_evals[-1] - cdf_evals[0]
        bin_probs = cdf_evals[1:] - cdf_evals[:-1]
        return bin_probs / z

    def transform_from_probs(probs: jax.Array) -> jax.Array:
        centers = (support[:-1] + support[1:]) / 2
        return jnp.sum(probs * centers)

    return transform_to_probs, transform_from_probs
```

Listing 2 An implementation of HL-Gauss (Imani & White, 2018) in PyTorch (Paszke et al., 2019).

```python
import torch
import torch.special
import torch.nn as nn
import torch.nn.functional as F


class HLGaussLoss(nn.Module):
    def __init__(self, min_value: float, max_value: float, num_bins: int, sigma: float):
        super().__init__()
        self.min_value = min_value
        self.max_value = max_value
        self.num_bins = num_bins
        self.sigma = sigma
        self.support = torch.linspace(
            min_value, max_value, num_bins + 1, dtype=torch.float32
        )

    def forward(self, logits: torch.Tensor, target: torch.Tensor) -> torch.Tensor:
        return F.cross_entropy(logits, self.transform_to_probs(target))

    def transform_to_probs(self, target: torch.Tensor) -> torch.Tensor:
        target = torch.clip(target, self.min_value, self.max_value)
        cdf_evals = torch.special.erf(
            (self.support - target.unsqueeze(-1))
            / (torch.sqrt(torch.tensor(2.0)) * self.sigma)
        )
        z = cdf_evals[..., -1] - cdf_evals[..., 0]
        bin_probs = cdf_evals[..., 1:] - cdf_evals[..., :-1]
        return bin_probs / z.unsqueeze(-1)

    def transform_from_probs(self, probs: torch.Tensor) -> torch.Tensor:
        centers = (self.support[:-1] + self.support[1:]) / 2
        return torch.sum(probs * centers, dim=-1)
```

C. Experimental Methodology

In the subsequent sections we outline the experimental methodology for each domain herein.

C.1. Atari

Both our online and offline RL regression baselines are built upon the Jax (Bradbury et al., 2018) implementation of DQN+Adam in Dopamine (Castro et al., 2018). Similarly, each of the classification methods (i.e., HL-Gauss and Two-Hot) were built upon the Jax (Bradbury et al., 2018) implementation of C51 in Dopamine (Castro et al., 2018). Hyperparameters for DQN+Adam are provided in Table C.1 along with any hyperparameter differences for C51 (Table C.2), Two-Hot (Table C.2), and HL-Gauss (Table C.3). Unless otherwise stated the online RL results in the paper were ran for 200M frames on 60 Atari games with five seeds per game. The offline RL results were ran on the 17 games in Kumar et al. (2021) with three seeds per game. The network architecture for both the online and offline results is the standard DQN Nature architecture that employs three convolutional layers followed by a single non-linear fully-connected layer before outputting the action-values.

Table C.1. **DQN+Adam Hyperparameters.**

Discount Factor γ	0.99
n-step	1
Minimum Replay History	$20,000$ agent steps
Agent Update Frequency	4 environment steps
Target Network Update Frequency	$8,000$ agent steps
Exploration ϵ	0.01
Exploration ϵ decay	$250,000$ agent steps
Optimizer	Adam
Learning Rate	6.25×10^{-5}
Adam ϵ	1.5×10^{-4}
Sticky Action Probability	0.25
Maximum Steps per Episode	$27,000$ agent steps
Replay Buffer Size	$1,000,000$
Batch Size	32

Table C.2. **C51 & Two-Hot Hyperparameters.** Difference in hyperparameters from DQN+Adam Table C.1.

Number of Locations	51
$[v_{\min}, v_{\max}]$	$[-10, 10]$
Learning Rate	0.00025
Adam ϵ	0.0003125

Table C.3. **HL-Gauss Hyperparameters.** Difference in hyperparameters from C51 Table C.2.

Smoothing Ratio σ/ς	0.75

C.1.1. Mixtures of Experts

All experiments ran with SoftMoE reused the experimental methodology of Obando-Ceron et al. (2024b). Specifically, we replace the penultimate layer of the DQN+Adam in Dopamine (Castro et al., 2018) with a SoftMoE (Puigcerver et al., 2024) module. The MoE results were ran with the Impala ResNet architecture (Espeholt et al., 2018). We reuse the same set of 20 games from Obando-Ceron et al. (2024b) and run each configuration for five seeds per game. All classification methods reused the parameters from Table C.2 for C51 and Two-Hot or Table C.3 for HL-Gauss.

C.1.2. Multi-Task & Multi-Game

The multi-task and multi-game results follow exactly the methodology outlined in Ali Taïga et al. (2023) and Kumar et al. (2023) respectively. We reuse the hyperparameters for HL-Gauss outlined in Table C.3. For multi-task results each agent is run for five seeds per game. Due to the prohibitive compute of the multi-game setup we run each configuration for one seed.

C.2. Wordle

Our Wordle experiments follow the methodology in Snell et al. (2023) by using a GPTlike decoder-only Transformer model trained using a standard autoregressive objective. We make use of the entire dataset of Wordle games compiled by Snell et al. (2023). In Figure 4 we report the success rate of guessing the word in one turn of play. See Figure C.1 for an illustration of the input and output of the Transformer model when playing a game of Wordle.

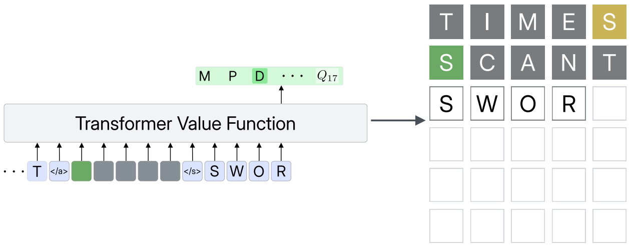

Figure C.1. **Illustration of a Transformer playing Wordle.** At a given timestep the Transformer model takes as input the board state along with the presence of letters in the hidden word. The model then predicts the next character to be played.

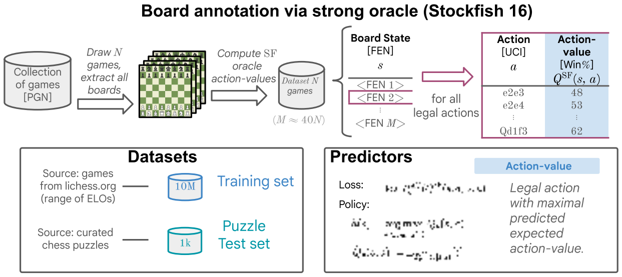
Board annotation via strong oracle (Stockfish 16)
Collection of games [PGN]
Draw N games, extract all boards
Compute SF oracle action-values
Dataset N games
$(M \approx 40N)$
Board State [FEN] s
<FEN 1>
<FEN 2>
<FEN M>
for all legal actions
Action [UCI] a
Action-value [Win%] $Q^{SF}(s, a)$
e2e3 48
e2e4 53
Qd1f3 62
Datasets
Source: games from lichess.org (range of ELOs)
10M
Training set
Source: curated chess puzzles
1k
Puzzle Test set
Predictors
Action-value
Loss:
Policy:
Legal action with maximal predicted expected action-value.

Figure C.2. **Dataset generation for Q-value distillation on Chess.** The Stockfish chess engine is used to annotate a collection of board states in the PGN format.

C.3. Chess

We follow exactly the setup in Ruoss et al. (2024) with the only difference being the use of HL-Gauss with a smoothing ratio $\sigma/\varsigma = 0.75$. Specifically, we take the action-values produced by Stockfish and project them a categorical distribution using HL-Gauss. As Ruoss et al. (2024) was already performing classification we reuse the parameters of their categorical distribution, those being, $m = 128$ bins evenly divided between the range $[0, 1]$. For each parameter configuration we train a single agent and report the evaluation puzzle accuracy. Puzzle accuracy numbers for one-hot and AlphaZero w/ MCTS were taken directly from Ruoss et al. (2024, Table 6).

C.4. Robotic Manipulation

We study a large-scale vision-based robotic manipulation setting on a mobile manipulator robot with 7 degrees of freedom, which is visualized in Figure 6 (left). The tabletop robot manipulation domain consists of a tabletop with various randomized objects spawned on top of the countertop. A RetinaGAN is applied to transform the simulation images closer to real-world image distributions, following the method in (Ho et al., 2021). We implement a Q-Transformer policy following the procedures in (Chebotar et al., 2023). Specifically, we incorporate autoregressive Q-learning by learning Q-values per action dimension, incorporate conservative regularization to effectively learn from suboptimal data, and utilize Monte-Carlo returns.

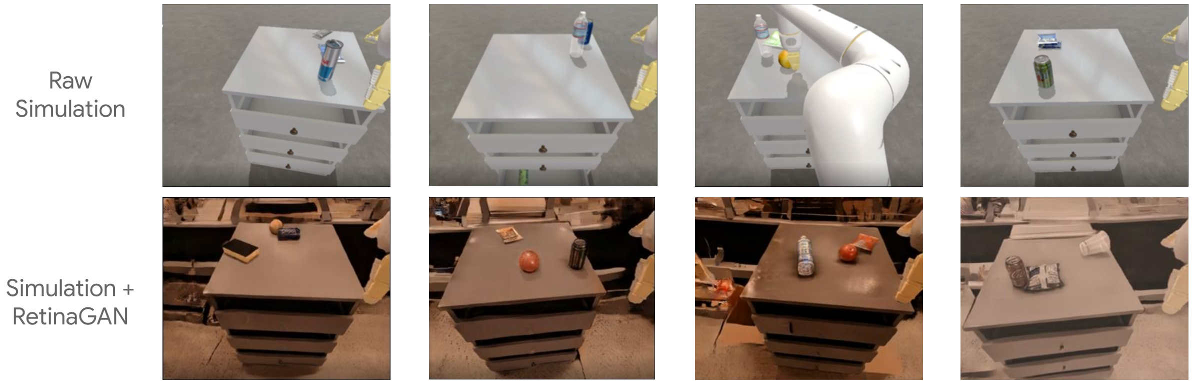

Figure C.3. **Robot manipulation domain.** The simulated robot manipulation (§4.3) consists of a tabletop with randomized objects. A learned RetinaGAN transformation is applied to make the visual observation inputs more realistic.

C.5. Regression Target Magnitude & Loss of Plasticity

To assess whether classification losses are more robust when learning non-stationary targets of increasing magnitude we leverage the synthetic setup from Lyle et al. (2024). Specifically, we train a convolutional neural network that takes CIFAR

10 images x_i as input and outputs a scalar prediction: $f_\theta : \mathbb{R}^{32\times32\times3} \to \mathbb{R}$. The goal is to fit the regression target,

$$y_i = \sin(m f_{\theta^-}(x_i)) + b$$

where $m = 10^5$, θ^- are a set of randomly sampled target parameters for the same convolutional architecture, and b is a bias that changes the magnitude of the prediction targets. It is clear that increasing b shouldn't result in a more challenging regression task.

When learning a value function with TD methods the regression targets are non-stationary and hopefully increasing in magnitude (corresponding to an improving policy). To simulate this setting we consider fitting the network f_θ on the increasing sequence $b \in \{0, 8, 16, 24, 32\}$. For each value b we sample a new set of target parameters θ^- and regress towards y_i for $5,000$ gradient steps with a batch size of 512 with the Adam optimizer using a learning rate of 10^{-3}. We evaluate the Mean-Squared Error (MSE) throughout training for three methods: Two-Hot, HL-Gauss, and L2 regression. For both Two-Hot and HL-Gauss we use a support of $[-40, 40]$ with 101 bins.

D. Additional Results

D.1. Scaling with Mixtures of Experts

Recently, Obando-Ceron et al. (2024b) demonstrate that while parameter scaling with convolutional networks hurts single-task RL performance on Atari, incorporating Mixture-of-Expert (MoE) modules in such networks improves performance. Following their setup, we replace the penultimate layer in the architecture employed by Impala (Espeholt et al., 2018) with a SoftMoE (Puigcerver et al., 2024) module and vary the number of experts in $\{1, 2, 4, 8\}$. Since each expert is a copy of the original penultimate layer, this layer's parameter count increases by a factor equal to the number of experts. The only change we make is to replace the MSE loss in SoftMoE DQN, as employed by Obando-Ceron et al. (2024b), with the HL-Gauss cross-entropy loss. We train on the same subset of 20 Atari games used by Obando-Ceron et al. (2024b) and report aggregate results over five seeds in Figure D.1.

As shown in Figure D.1, we find that HL-Gauss consistently improves performance over MSE by a constant factor independent of the number of experts. One can also observe that SoftMoE + MSE seems to mitigate some of the negative scaling effects observed with MSE alone. As SoftMoE + MSE uses a softmax in the penultimate layer this could be providing similar benefits to using a classification loss but as we will later see these benefits alone cannot be explained by the addition of the softmax.

D.2. Replay Ratio Scaling

In supervised learning label smoothing is thought to reduce overfitting. To assess whether HL-Gauss achieves a similar effect, we utilize a common deep RL paradigm of increasing the number of updates per environment step, known as the replay ratio (D'Oro et al., 2023). A high replay ratio makes the agent more susceptible to overfitting on early experiences,

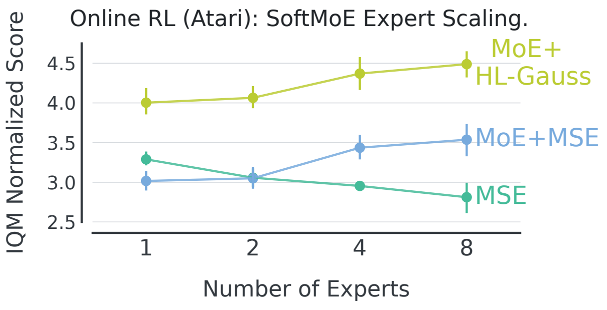

Figure D.1. **MoE scaling curves for HL-Gauss and MSE on Online RL**. HL-Gauss, with a single expert, outperform all regression configurations. Both HL-Gauss and MSE scale similarly when employing SoftMoE, with HL-Gauss providing $\approx 30\%$ IQM improvement. SoftMoE also mitigates negative scaling observed with MSE alone. See §D.1 for more details.

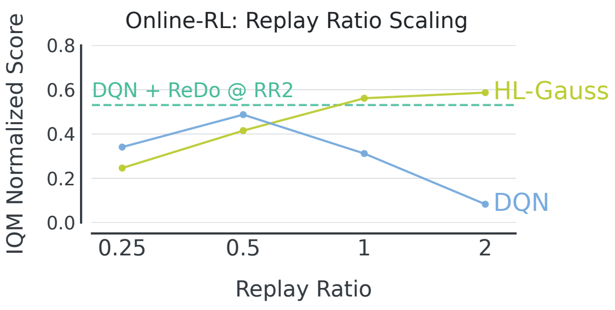

Figure D.2. **Replay Ratio Scaling on Atari.** Results for HL-Gauss when scaling the replay ratio, that is, the number of gradient steps per environment step. Plotted is the IQM after 10M frames over 17 games ran with 3 seeds per game following the setup in Sokar et al. (2023). HL-Gauss shows positive scaling with replay ratio even outperforming ReDo (Sokar et al., 2023) which itself improves upon resets (Nikishin et al., 2022; D'Oro et al., 2023) in Atari.

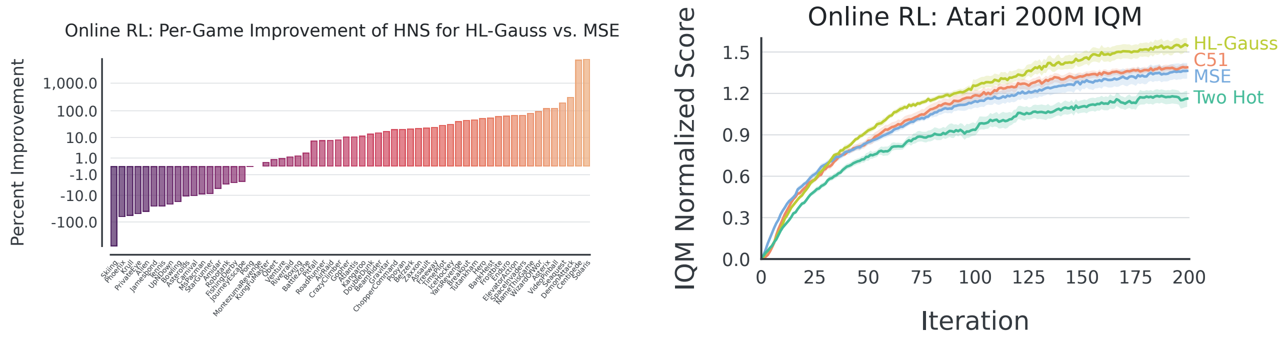

Figure D.3. **HL-Gauss vs MSE per game in single-task online RL (§4.2). (Left)** Each column displays the relative final performance of HL-Gauss with respect to MSE in the single-task online RL training curves. This is a summary of the curves displayed in Figure D.7. Note that HL-Gauss outperforms MSE in $\approx 3/4$ of all games, and that HL-Gauss scores at least 10% higher on 1/2 of all games. **(Right)** IQM normalized training curves throughout training.

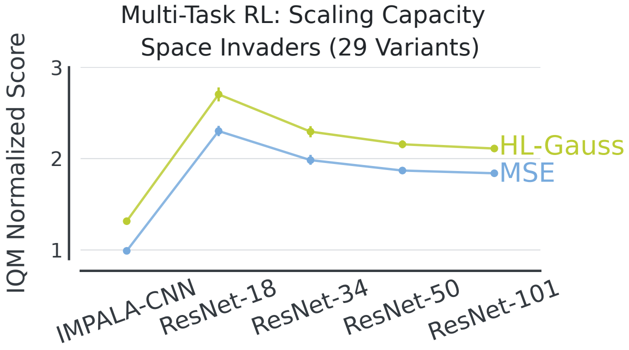

Figure D.4. **Multi-task Online RL**. Online RL scaling results with actor-critic IMPALA with ResNets on SPACE INVADERS. HL-Gauss outperforms MSE for all models. Since human scores are not available for variants, we report normalized scores using a baseline IMPALA agent with MSE loss. See §4.2 for more details.

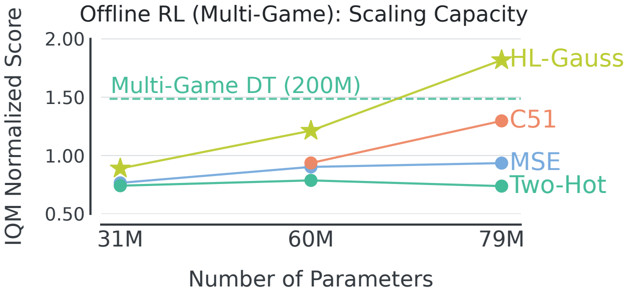

Figure D.5. **Multi-task Offline RL results presented in terms of DQN normalized scores**. Note that when aggregate results are computed with DQN normalization, HL-Gauss exhibits a faster rate of improvement than C51 as the number of parameters scales up.

leading to an effect known as the primacy bias (Nikishin et al., 2022). To test this hypothesis, we follow the setup from Sokar et al. (2023) and train DQN (Adam+MSE) (Obando-Ceron & Castro, 2021; Mnih et al., 2015) and HL-Gauss (Imani & White, 2018) for 10M frames on 17 Atari 2600 games in Sokar et al. (2023). We increase the replay ratio from the default value of 0.25 in DQN to 2 (8x higher), specifically testing replay ratios $0.25, 0.5, 1, 2$.

Figure D.2 presents these results averaged over 3 seeds. The results show that DQN with MSE degrades substantially as we increase the update ratio. However, HL-Gauss seems to reduce overfitting and better maintain plasticity, as evidenced by positive scaling with replay ratio. Notably, HL-Gauss performs equivalently to ReDo (Sokar et al., 2023), which claims to improve upon methods like hard network resets (Nikishin et al., 2022).

D.3. Per-Game Atari Results

Figure D.7 presents per-game training curves for DQN (Adam+MSE) (Obando-Ceron & Castro, 2021; Mnih et al., 2015), Two-Hot (Schrittwieser et al., 2020), C51 (Bellemare et al., 2017), and HL-Gauss (Imani & White, 2018; Imani et al., 2024). Each agent is trained for 200M frames with one iteration corresponding to 1M frames. Results are reported over 5 seeds with 95% confidence intervals are represented as the shaded region. The left subplot of Figure D.3 provides per-game improvements of HL-Gauss over MSE and the right subplot shows IQM results throughout training.

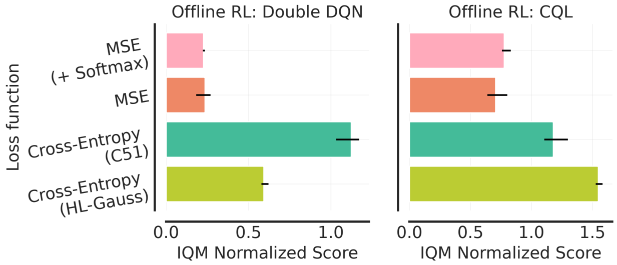

Figure D.6. Evaluating learning stability from softmax in offline RL. We do not observe any gains from using a softmax operator with the MSE loss.

D.4. Multi-Task Atari Results

Figure D.4 shows aggregate results for 29 variants of Space Invaders across different network architectures. Notably, HL-Gauss only provides a constant performance improvement over MSE across all architectures. This result is consistent with Ali Taïga et al. (2023) and we hypothesize this is primarily due to the amount and diversity of pre-training data. Notably, in Asteroids there are twice as many pre-training variants as compared to Space Invaders, which may result in substantially reduced diversity in the case of Space Invaders. Additionally, Figure D.5 presents results for multi-game setup (§4.2) with DQN-normalized scores. Notably, we see HL-Gauss outpace C51 when scaling parameters with HL-Gauss also substantially outperforming the multi-Game decision transformers (Lee et al., 2022) at 79M parameters.

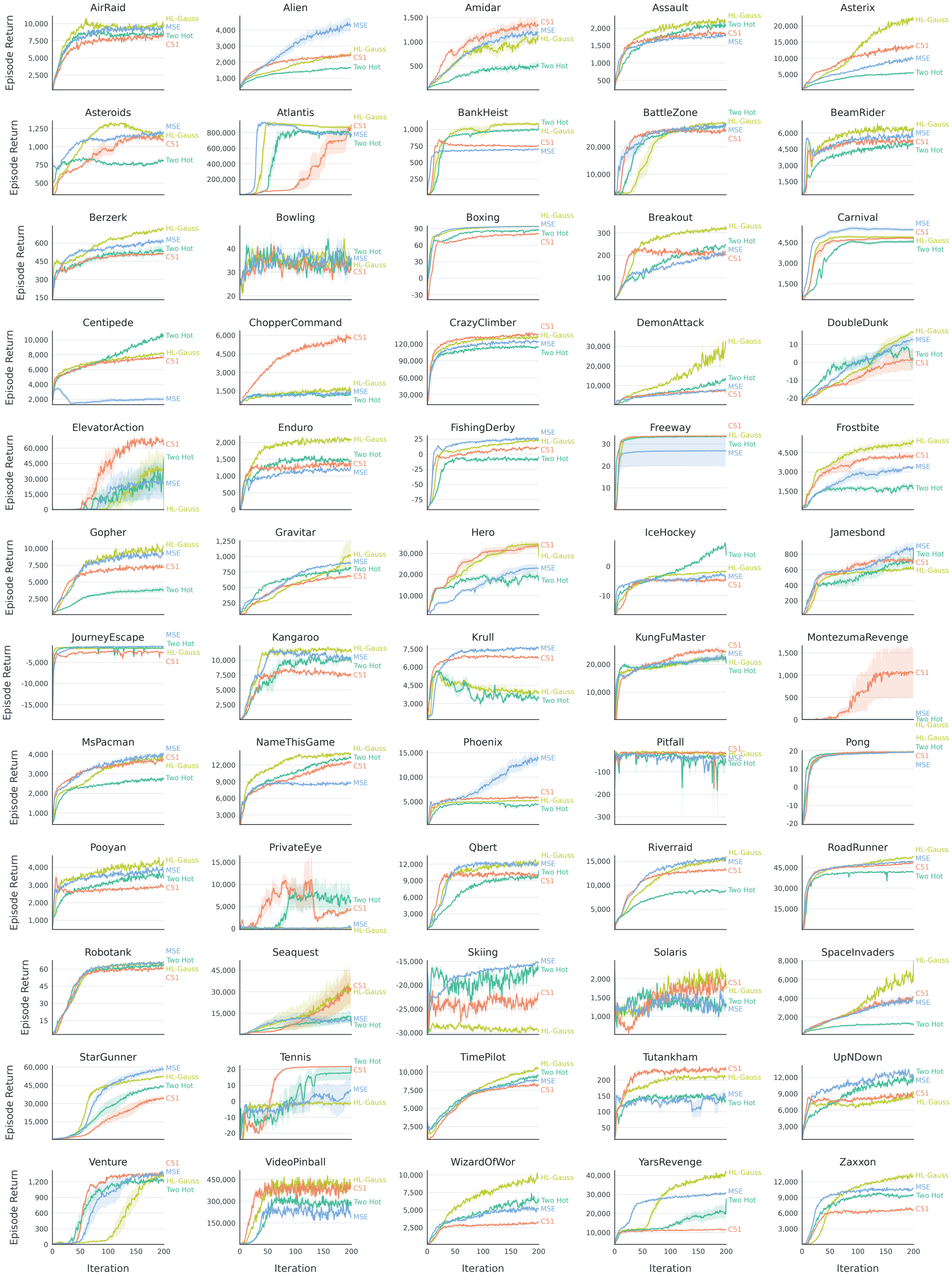

Figure D.7. **Training curves on single-task online RL (§4.1) for all 60 Atari games.** All games ran for 200M frames and ran for: DQN(Adam), C51, Two-Hot, and HL-Gauss.