LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR MOBILE UND VERTEILTE SYSTEME

Trends in Mobilen und Verteilten Systemen

# Curiosity und Diversity im Kontext von Reinforcement Learning - Ein Vergleich

Bearbeiter: David Jonathan Müller
Lukas Johannes Rieger

Betreuer: M. Sc. Thomas Gabor

Aufgabensteller: Prof. Dr. Claudia Linnhoff-Popien

Dokument erstellt
21. Juni 2020

# Curiosity und Diversity im Kontext von Reinforcement Learning - Ein Vergleich

**Zusammenfassung**

Dieses Dokument ist die Vorlage für Seminararbeiten am Lehrstuhl für Mobile und Verteilte Systeme.

**Inhaltsverzeichnis**

# 1 Einleitung

Das wohl herausstechendste Merkmal des Menschen ist seine Fähigkeit, komplexe Probleme in unterschiedlichsten Umgebungen und oftmals ohne vorheriges Wissen zu lösen. Diese Fähigkeit ist nicht nur für organische Lebewesen, sondern auch für künstliche Aktoren in einer Vielzahl an Szenarien äußerst nützlich. Ein häufig verwendeter Ansatz im Bereich der künstlichen Intelligenz stellt das "Reinforcement Learning" dar. Dieser Ansatz erlaubt es Agenten, diverse Aktionen im Kontext ihrer jeweiligen Umgebung zu erlernen und zu verbessern, ohne dabei notwendigerweise durch eine dritte Instanz überwacht werden zu müssen.[**?**] Ein zentraler Problembereich stellt hierbei das Finden und Konfigurieren von "Belohnungen", die dem Agenten, ähnlich wie beim Menschen, Rückmeldung über die von ihm ausgeführten Aktionen geben soll.

Da in den meisten realen Anwendungsfällen externe Belohnungen nur spärlich oder überhaupt nicht vorhanden sind, benötigt man zusätzliche Methoden, um einen Lernfortschritt zu erzielen.[**?**] Intuitiv lässt sich ein Lernanreiz entweder als extrinsisch, oder als intrinsisch definieren. Dient der Lernvorgang dem erreichen eines externen, vorgegebenen Ziels, lässt sich dieser als extrinsisch kategorisieren. In diesem Falle fällt es ebenfalls leicht, die Belohnung eines Aktoren in Abhängigkeit seines Zielfortschritts zu definieren. Da in realistischen Szenarien eine solche direkte Bewertung meist nur schwer vorzunehmen ist, muss eine etwaige Belohnung ebenfalls intrinsische Anreize in Betracht ziehen. Von solchen intrinsischen Anreizen spricht man dann, wenn die zugrundeliegende Aktion *inhärent* sinnvoll erscheint, also nicht von externen Gegebenheiten abhängt.

So scheint es etwa für den Menschen intuitiv sinnvoll, in einer unbekannten Umgebung möglichst viele unterschiedliche Vorgehensweisen auszuprobieren und sich dabei auch Strategien zu merken, welche vielleicht erst zu einem späteren Zeitpunkt Anwendung finden. Ein solches Vorgehen wäre auch bei einer künstlichen Intelligenz wünschenswert. Zu diesem Zweck existieren unterschiedliche Ansätze, welche sich allerdings in gewissen Punkten durchaus ähnlich sind.

In dieser Arbeit betrachten und vergleichen wir zwei Methodiken, welche das klassische Reinforcement Learning in dieser Hinsicht erweitern. Zu diesem Zweck wird zunächst der Begriff Reinforcement Learning erklärt. Die anschließenden Kapitel fokussieren sich auf zwei Ansätze zur Bewertung des Lernfortschritts. In diesem Rahmen stellen wir zuerst den Ansatz der "Curiosity" nach [**?**] vor. Wir betrachten außerdem das Konzept der "Diversity" und dessen Anwendung im Reinforcement Learning. Diese stellen wir daraufhin vergleichend gegenüber und gehen auf deren Gemeinsamkeiten und Kernunterschiede ein.

# 2 Grundlagen

## 2.1 Reinforcement Learning im Allgemeinen

## 2.2 Markov Entscheidungsprozesse

## 2.3 Bayes'sche Inferenz

# 3 Das Curiosity Modell nach Schmidhuber

## 3.1 Grundlagen des Curiosity Modells

### 3.1.1 Grundlegende Prinzipien

Um Schmidhubers System verstehen zu können, ist es wichtig, zuerst einige zentrale Grundbegriffe zu definieren. Die Säulen seines Werks bilden vor allem vier zentrale Prinzipien, die in $curiosity_schmidhuberdargelegtwerden$ :

**Information ist "heilig"** Diesem Prinzip zufolge, muss die gesamte Abfolge an Aktionen und Beobachtungen eines Aktoren gespeichert werden. Begründet wird dies auf der Basis, dass diese Informationen die absolute Basis für jegliches Wissen darstellt, das der Aktor über seine Umgebung besitzt. Grundlegend für dieses Prinzip ist hierbei die Annahme, dass es überhaupt realistisch *möglich* ist, einen solchen Katalog aller Daten physisch zu speichern. Schmidhuber argumentiert, dass dies durchaus nicht unrealistisch sei und zeigt dies mit Hilfe eines Beispiels. Für dieses sind zuerst einige Annahmen nötig:

- Ein menschliches Leben dauert im Durchschnitt nicht länger als $3 \times 10^9$ Sekunden an.
- Ein menschliches Gehirn verfügt in etwa über $10^{10}$ Neuronen, welche jeweils über ca. $10^4$ Synapsen verfügen.

Nimmt man nun an, dass lediglich die halbe Hirnkapazität genutzt wird, um rohe Daten zu speichern und jede Synapse höchstens 6 bit speichern kann, so ist es durchaus möglich, die gesamten sensorischen Eindrücke eines Lebens bei einer Rate von $10^5\, bits/s$ zu sichern.

**Verbessern von subjektiver Komprimierbarkeit** Demzufolge lässt sich jede Regularität im Informationsstrom dazu nutzen, diesen weiter zu komprimieren. Eine solche komprimierte Version des ursprünglichen Datenstroms lässt sich also als eine Art *vereinfachter Erklärung* von diesem interpretieren. Daraus folgt, dass ein Agent zumindest einen Teil seiner Rechenzeit darauf verwenden sollte, mithilfe eines Kompressionsalgorithmus seine Daten zu komprimieren.

**Intrinsische Belohnungen spiegeln Kompressionsfortschritt wieder** Ein Agent sollte seinen Kompressionsfortschritt überwachen und bei erfolgreicher Komprimierung entsprechend darauf reagieren. In Abhängigkeit der eingesparten Bits wird dementsprechend eine intrinsische Belohnung für den Agenten generiert.

**Intrinsische Belohnungen maximieren** Ein entsprechender Reinforcement Learning Algorithmus kann nun versuchen, die zu erwartende intrinsische Belohnung zu maximieren. Laut Schmidhuber würde sich ein solcher Algorithmus vor allem auf solche Aktionen fokussieren, die es erlauben, neue aber erlernbare Regularitäten zu finden oder zu erstellen.

### 3.1.2 Interne Symbole als Konsequenz effizienter Komprimierung

Schmidhuber beschreibt weiter, aufbauend auf einer generellen Historie an Beobachtungen, wie ein vorausschauendes neurales Netz, das als Kompressor fungiert, spezielle interne Repräsentierungen generiert, welche über häufig auftretende Dinge abstrahieren. Diese internen Darstellungen werden kurz *Symbole* genannt [**?**, p. 6]. Ein naheliegendes Beispiel findet sich laut Schmidhuber etwa im Tag/Nacht-Rhythmus. Da der Sonnen Auf- und Untergang sich beständig wiederholt, ist es effizienter, ein spezifisches Symbol für diesen Prozess zu generieren und die Komprimierung dadurch voranzutreiben.

**TODO: Reicht das ? Also, Consciousness**

## 3.2 Curiosity im Kontext von Historienkomprimierung

**Subjektive Interessantheit als Ableitung subjektiver Schönheit** Laut Schmidhuber lässt sich die subjektive Schönheit einer Beobachtung als die Menge an benötigten Bits definieren, um diese Beobachtung zu kodieren [**?**, p. 7] Genauer formuliert bedeutet dies:
Sei $O(t)$ der Zustand eines subjektiven Beobachters am Zeitpunkt $t$. Die subjektive Schönheit sei gegeben durch $B(D, O(t))$, wobei es sich bei $D$ um die jeweilige Beobachtung handelt. Dieses Maß der subjektiven Schönheit ist proportional zur Menge an Bits um $D$ zu beschreiben, wobei jegliches vorheriges Wissen des Beobachters ebenfalls in Betracht gezogen werden muss. [**?**, p. 7] Schmidhuber erklärt diese Definition anhand des Beispiels eines menschlichen Gesichts. So würde es sich für einen Kompressor anbieten, eine interne Repräsentation eines archetypischen Gesichts zu generieren, um bereits beobachtete Gesichter möglichst effizient zu kodieren. Die Beobachtung eines neuen Gesichts erlaubt es dann, lediglich die spezifischen Abweichungen dieses neuen Gesichts vom internen Prototypen zu speichern. Ein solcher Beobachter würde also genau solche Gesichter als subjektiv am schönsten klassifizieren, die am geringsten vom internen Gesichtsprototypen des Beobachters abweichen. [**?**, p. 7]

Schmidhuber führt nun, aufbauend auf der subjektiven Schönheit, den Begriff der *subjektiven Interessantheit* ein. Er argumentiert, dass eine subjektiv schöne Beobachtung nur so lange interessant ist, bis der Beobachter die spezifische Regularität des Objekts komplett verarbeitet hat.[**?**, p. 7-8] In anderen Worten ist eine Beobachtung also nur dann für einen Beobachter interessant, wenn dessen Kompressor diese Beobachtung noch nicht optimal zusammenfassen kann.

Genauer definiert Schmidhuber nun die *zeitabhängige* subjektive Interessantheit $I(D, O(t))$ einer Beobachtung $D$ in Relation zu einem Beobachter $O$ am Zeitpunkt $t$ als

$$I(D, O(t)) \sim \frac{\partial B(D, O(t))}{\partial t} \tag{1}$$

, also der ersten Ableitung der subjektiven Schönheit [**?**, p. 8]. Ein Agent kann seine Beobachtungen also über Zeit besser analysieren und etwaige Wiederholungen oder Regularitäten erkennen, wodurch die beobachteten Daten subjektiv schöner werden. Solange diese Komprimierungsprozess anhält, handelt es sich um *interessante* Daten. [**?**, p. 8]

#### 3.2.1 Wie neugierige Agenten operieren

Auf Basis der subjektiven Schönheit und darauf aufbauend, der subjektiven Interessantheit, lässt sich nun erläutern, wie etwa ein auf Reinforcement Learning basierender Agent handeln würde. Schmidhuber setzt an, dass im Falle von fehlenden äußeren Belohnungen, ein entsprechender Agent versuchen würde, die *Interessantheit* zu maximieren [**?**, p. 8]. Es werden also genau solche Sequenzen an Aktionen vom Agenten ausgewählt, welche zukünftig den zu erwartenden Kompressionsfortschritt maximieren [**?**, p. 8]

### 3.3 Kompressoren und Agenten im Detail

#### 3.3.1 Belohnungsmaximierung und Bewertung des Kompressorfortschritts

Die Abschnitte **??** und **??** fokussieren sich hauptsächlich auf die theoretischen Konzepte der *Komprimierung* von Daten und davon ausgehend, die Konzepte der *subjektiven Schönheit* und *subjektiven Interessantheit*. Die folgende Sektion befasst sich im Gegenzug mit Schmidhubers formalisierten Implementierungen dieser Konzepte.

Betrachtet wird zuerst ein Agent, der seine Umgebung in konkreten Zeitabschnitten $t = 1, 2, \ldots, T$ wahrnimmt und verändert. In den folgenden Abschnitten werden, Schmidhubers Schreibweise folgend, zeitabhängige Variablen $Q$ zum Zeitpunkt $t$ durch $Q(t)$, die geordnete Sequenz an Werten $Q(1), \ldots, Q(t)$ durch $Q(\leq t)$ und die Sequenz $Q(1), \ldots, Q(t-1)$ durch $Q(< t)$ dargestellt. Nimmt man an, dass der Agent zu jedem beliebigen Zeitpunkt $t$ einen realen Eingabewert $x(t)$ von der Umgebung erhält und daraufhin eine reale Aktion $y(t)$ ausführt, so ergibt sich als finales Ziel für den Agenten die Maximierung des zukünftigen *Nutzen*

$$u(t) = E_\mu \left[ \sum_{\tau=t+1}^{T} r(\tau) \mid h(\leq t) \right] \tag{2}$$

wobei $r(t)$ einen zusätzlichen realen Belohnungswert, $h(t)$ das geordnete Tripel $[x(t), y(t), r(t)]$ und $E_\mu(\cdot \mid \cdot)$ den bedingten Erwartungsoperator in Bezug auf eine möglicherweise unbekannte Verteilung $\mu$ aus einer Menge $M$ möglicher Verteilungen darstellt. [**?**, p. 17] In diesem Kontext stellt $h(t)$ also die Historie bis zu dem Zeitpunkt $t$ dar.

**Leistungsmessung eines Kompressors** Zur Bewertung eines Kompressors $p$ der eine Historie $h(\leq t)$ zum Zeitpunkt $t$ komprimiert, führt Schmidhuber den

Wert

$$C_l(p, h(\leq t)) = l(p) \tag{3}$$

ein. In diesem Kontext stellt $l(p)$ die Länge von $p$ in Bits dar. Je kürzer der Kompressor also ist, desto mehr Regelmäßigkeit lässt sich in den bisherigen Beobachtungen finden. [**?**, p. 19] Das maximale Limit von $C_l(p, h(\leq t))$ stellt laut Schmidhuber die Kolmogorov Komplexität $K^*(h(\leq t))$, also das kürzeste Programm, welches eine Ausgabe produziert, die mit $h(\leq t)$ beginnt, dar.

Während eine Leistungsmessung dieser zwar Art möglich ist, muss idealerweise auch die *Zeit* in Betracht gezogen werden, die ein Kompressor benötigt, um eine gegebene Historie $h$ zu komprimieren. Andernfalls wäre ein Szenario denkbar, in der ein Kompressor zwar eine extrem kompakte Repräsentation der vorliegenden Daten produzieren könnte, für diesen Prozess aber extrem viel Rechenzeit benötigt und somit praktisch unbrauchbar wäre. Aus diesem Grund stellt Schmidhuber noch einen zweiten Messungsansatz vor, welcher eine Reduzierung der Rechenzeit um $\frac{1}{2}$ äquivalent zu einer Komprimierung um 1 Bit behandelt [**?**, p. 19]

$$C_{l\tau}(p, h(\leq t)) = l(p) + \log \tau(p, h(\leq t)) \tag{4}$$

**Bewertung des Kompressor Fortschritts** Die Leistung eines Kompressors an sich stellt allerdings nicht den Kernaspekt des Curiosity-Ansatzes dar. Ähnlich wie schon im Abschnit **??** liegt der Fokus viel mehr auf der *Änderung* dieses Werts über mehrere Zeitschritte hinweg - also der *Verbesserung* des Kompressors. Dementsprechend definiert Schmidhuber den internen Belohnungswert in Reaktion auf den Fortschritt des Kompressors als

$$r_{int}(t+1) = f\left[C(p(t), h(\leq t+1)), C(p(t+1), h(\leq t+1))\right] \tag{5}$$

wobei $f$ jeweils reale Paare auf selbige abbildet. [**?**, p. 19]

**TODO: Was bedeutet das? Beispiel für f nennen. (f(a,b) = a - b)**

**TODO: A.5 ist super wichtig. Da erklärt er genau wie sich der interne Reward ergibt.**

### 3.3.2 Auswahl von zielführenden Aktionen

# 4 Diversity im Reinforcement Learning

In diesem Kapitel beschäftigen wir uns damit, wie das Konzept von Diversity im Bereich des Reinforcement Learning (RL) Anwendung finden kann. Ziel ist, dass ein RL Agent in einer unüberwachten Phase zunächst Fähigkeiten erlernt, welche das Bewältigen von Aufgaben in der darauf folgenden, überwachten Phase erleichtern sollen.

Dieses Kapitel stützt sich zu einem Großteil auf die wissenschaftliche Arbeit *Diversity is all you need: Learning skills without a reward function*[**?**]. Falls nicht anders angegeben, wurden die Informationen hieraus entnommen.

## 4.1 Klärung wichtiger Begriffe aus der Informationstheorie

Im Folgenden werden Begriffe aus der Informationstheorie verwendet, welche es zunächst zu klären gilt. Wir betrachten *Entropie* und *Transinformation* nach [**?**] und [**?**].

Die *Entropie* beschreibt in der Informationstheorie den mittleren Informationsgehalt bzw. die Ungewissheit einer Quelle. Ist beispielsweise bei einer Ereignismenge jedes Ereignis gleich wahrscheinlich, so ist die Ungewissheit maximal [**?**].

Nach [**?**] besitzt eine diskrete Zufallsvariable $X$ mit dem Zeichenvorrat $\mathcal{X}$ und der Wahrscheinlichkeitsfunktion $p(x)$ die *Entropie*

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \cdot log\ p(x)$$

Diese lässt sich nach [**?**] auch über den Erwartungswert $E$ berechnen unter Verwendung von

$$H(X) = E_p\ log\ \frac{1}{p(X)}$$

, woraus unter Anwendung der Rechenregeln für Logarithmus und Erwartungswert trivial

$$H(X) = -E_p\ log\ p(X) \tag{6}$$

folgt.

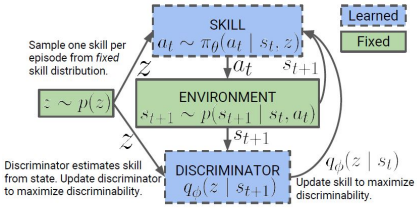
Desweiteren ist die Formel für die *bedingte Entropie* nach [**?**] gegeben durch

$$H(Y|X) = -E_{p(x,y)}\ log\ p(Y|X) \tag{7}$$

Die *Transinformation* beschreibt die Menge an Information, die eine Zufallsvariable über eine andere enthält bzw. die Reduktion der Ungewissheit aufgrund des Wissens um die jeweils andere Zufallsvariable [**?**].

Mathematisch wichtig für uns ist lediglich der Zusammenhang zwischen *Transinformation* und *Entropie*, welcher nach [**?**] gegeben ist durch

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{8}$$




**Algorithm 1:** DIAYN

**while** *not converged* **do**
  Sample skill $z \sim p(z)$ and initial state $s_0 \sim p_0(s)$
  **for** $t \leftarrow 1$ **to** *steps_per_episode* **do**
    Sample action $a_t \sim \pi_\theta(a_t \mid s_t, z)$ from skill.
    Step environment: $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$.
    Compute $q_\phi(z \mid s_{t+1})$ with discriminator.
    Set skill reward $r_t = \log q_\phi(z \mid s_{t+1}) - \log p(z)$
    Update policy ($\theta$) to maximize $r_t$ with SAC.
    Update discriminator ($\phi$) with SGD.

Abbildung 1: “Diversity is all you need” Algorithmus

Quelle: [**?**]

## 4.2 Funktionsweise

Das selbstständige Erlernen von brauchbaren Fähigkeiten, so wie in [**?**] beschrieben, baut auf drei Grundideen auf.

(1) Unterschiedliche Fähigkeiten sollten andere Zustände besuchen, sodass ihre Unterscheidbarkeit gewährleistet ist.

(2) Um besagte Fähigkeiten zu unterscheiden, werden nicht die Aktionen, sondern die Zustände betrachtet. Das liegt daran, dass Aktionen, welche die Umgebung nicht beeinflussen, für einen Beobachter nicht erkennbar sind. Das Paper verdeutlicht dies mit einem treffenden Beispiel. Für einen außenstehenden Betrachter ist nicht ersichtlich, wie fest ein Roboterarm eine Tasse in der Hand hält, falls sich diese nicht bewegt.

(3) Schließlich soll erreicht werden, dass die Fähigkeiten so vielfältig (*diverse*) wie möglich sind. Die Idee ist, dass unterscheidbare Fähigkeiten mit einer hohen Entropie einen Zustandsraum abdecken, welcher sich weit entfernt von anderen Fähigkeiten befindet.

Um diese Punkte zu realisieren, definieren wir zunächst die Zufallsvariablen $S$ und $A$ für Zustände und Aktionen. Sei nun $Z \sim p(z)$ eine latente Variable, unter deren Bedingung die Strategie definiert wird. Bei fixem $Z$ sprechen wir hier von einer ”Fähigkeit”. Entropie sowie Transinformation werden zur Basis $e$ berechnet.

Nun soll die Transinformation zwischen Fähigkeiten und Zuständen, $I(S; Z)$, maximiert werden um sicherzustellen, dass die Fähigkeit die vom Agent durchlaufenen Zustände vorgibt. Anders gesagt wird sichergestellt, dass aus den besuchten Zuständen auf die Fähigkeit geschlossen werden kann.

Außerdem minimieren wir die Transinformation zwischen Fähigkeit und Aktionen bei gegebenen Zustand, $I(A; Z|S)$. So wird garantiert, dass nicht Aktionen, sondern Zustände für die Unterscheidung von Fähigkeiten verwendet werden.

Maximiert wird auch die Entropie $H(A|S)$.

Insgesamt maximieren wir nach [?] also

$$\begin{aligned}\mathcal{F}(\theta) &\triangleq I(S;Z) + H(A|S) - I(A;Z|S) \qquad (9)\\ &= (H(Z) - H(Z|S)) + H(A|S) - (H(A|S) - H(A|S,Z))\\ &= H(Z) - H(Z|S) + H(A|S,Z) \qquad (10)\end{aligned}$$

Der Term wurde unter Verwendung von (??) umgeformt.

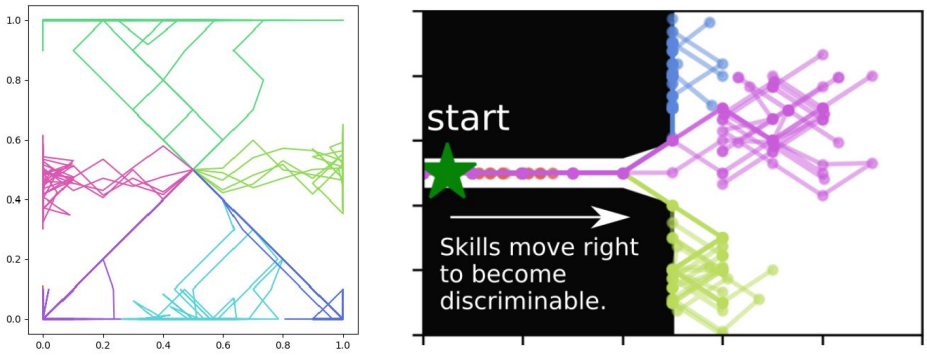
Da sich $p(z|S)$ nicht genau berechnen lässt, wird das Folgende mit einem discriminator $q_\phi$ approximiert. Mit der Jensenschen Ungleichung haben wir laut [?] eine untere Schranke $\mathcal{G}(\theta)$ für $\mathcal{F}(\theta, \phi)$:

$$\begin{aligned}\mathcal{F}(\theta) &= H(A|S,Z) - H(Z|S) + H(Z)\\ &= H(A|S,Z) + E_{z\sim p(z), s\sim\pi(z)}(log\ p(z|s)) - E_{z\sim p(z)}(log\ p(z)) \qquad (11)\\ &\geq H(A|S,Z) + E_{z\sim p(z), s\sim\pi(z)}(log\ q_\phi(z|s) - log\ p(z)) \triangleq \mathcal{G}(\theta,\phi)\end{aligned}$$

In (??) wurden die Entropien nach (??) und (??) umgeformt.

Nach der unüberwachten Trainingsphase verfügt der Agent über eine Sammlung von verschiedenartigsten Fähigkeiten, von denen vermutlich einige nutzlos sind. Aufgrund der implementierten Diversity müssen jedoch Fähigkeiten existieren, welche sich stark von den unbrauchbaren unterscheiden. So ist es intuitiv einleuchtend, dass es sich hierbei um sinnvolle Fähigkeiten handelt.

## 4.3 Experimente und Beispiele




(a) Zweidimensionale Navigation

(b) Überlagernde Fähigkeiten

Abbildung 2: TODO

Quelle: [?]

Wie bereits mehrfach angemerkt ist das Ziel das eigenständige Erlernen von vielfältigen Fähigkeiten. [?] liefert zunächst das simple Beispiel einer zweidimen-

sionalen Navigation in einem leeren Raum **??**. Hier ist sehr anschaulich zu erkennen, wie die farblich kodierten Fähigkeiten unterschiedliche Zustandsräume abdecken.

Einen Schritt weiter gedacht zeigt das Beispiel **??**, dass sich Fähigkeiten unter Umständen auch zeitweilig überlagern können, solange sie schlussendlich unterscheidbare Zustandsräume abdecken.

In beiden Beispielen ist gut zu erkennen, wie sich die Fähigkeiten gegenseitig "abstoßen" und so automatisch ein Großteil des Zustandsraums abgedeckt wird. Die erkundeten Zustände liegen außerdem gleichmäßig über dem Zustandsraum verteilt.

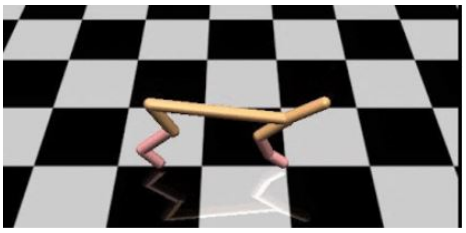


Abbildung 3: Experiment "Half Cheetah"

Quelle: [**?**]

Interessanter wird es bei den komplexeren Versuchsaufbauten. Wir betrachten im Folgenden den vom Paper [**?**] als "Half Cheetah" betitelten Agenten. Hierbei handelt es sich um ein hundartiges, zweidimensionales Wesen, welches die Kontrolle über sein Vorder- und Hinterbein besitzt (siehe **??**).

Ohne jegliche externe Belohnung hat dieser Agent gelernt, nach vorne und hinten zu rennen beziehungsweise zu gehen. Außerdem besitzt er die Fähigkeit, einen Vorwärtssalto zu machen. Wir empfehlen zu besseren Veranschaulichung die Einsicht der Videos auf [**?**].

Eine Reward-Funktion, welche ein solches Verhalten hervorruft, manuell zu definieren ist nicht einfach.

## 4.4 Verwendung der gelernten Fähigkeiten

Nachdem nun eine Sammlung von Fähigkeiten erlernt wurde stellt sich nun die Frage, wie sich diese effektiv nutzen lässt.

[**?**] führt hier als erstes an, dass der DIAYN Algorithmus als ein Vortraining genutzt werden kann. Danach extrahiert man die Fähigkeit mit dem größten Reward und passt entsprechend die Initialgewichte der eigentlichen Policy an. **??** zeigt für das Beispiel des Half Cheetahs, dass sich bei Initialisierung mit einre vorgelernten Fähigkeit (blaue Linie) schneller größere Rewards erzielen lassen als bei einer zufälligen Initialisierung (orangene Linie).

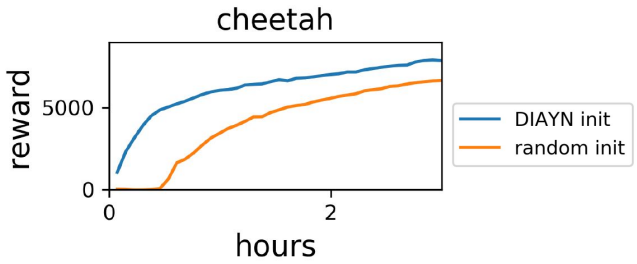
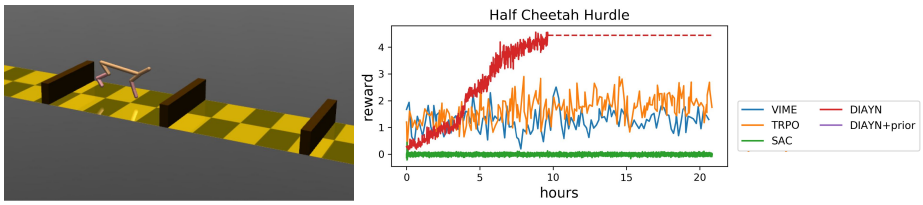



Abbildung 4: Policy Initialisierung

Quelle: [**?**]

Für uns interessanter ist allerdings die Verwendung für Hierarchisches Reinforcement Learning (HRL).

Nach [**?**] lernen HRL Methoden eine Policy, welche aus mehreren Schichten besteht. Jede Ebene kontrolliert hierbei ein andere temporäre Abstraktionsebene. Dies ermöglicht es dem Agenten, nicht nur Basisoperationen auszuführen, sondern auch komplexere Aktionen wie zum Beispiel Sequenzen von Operationen einer niedrigeren Ebene.




(a) (b)

Abbildung 5: DIAYN für HRL

Quelle: [**?**]

Laut [**?**] eignet sich DIAYN hervorragend als Grundstein für HRL. Für ein Experiment diesbezüglich betrachten wir wieder den Half Cheetah. Dieser hat nun die Aufgabe, über Hürden zu springen (siehe **??**) und erhält Rewards für deren überwinden.

Um die gelernten Fähigkeiten für HRL zu benutzen, wird DIAYN um einen "meta-controller" erweitert welcher kontrolliert, welche 10 Fähigkeiten jeweils als nächstes ausgeführt werden.

Das Experiment wird außerdem mit aktuellen Reinforcement Learning Algorithmen (VIME, TRPO und SAC) durchgeführt.

Wie die Grafik **??** zeigt, schneiden diese im Experiment sehr schlecht ab und es gibt quasi keine merkliche Steigerung des Rewards. Im Gegensatz dazu

zeigt der Ansatz mit dem DIAYN Algorithmus schnell deutliche Forschritte und übertrifft merklich die anderen.

Aufgrund der Knappheit von externen Rewards gestaltet sich diese Aufgabe extrem schwierig für traditionelle RL Algorithmen. Diese müssen quasi zufällig eine Hürde übersteigen, um eine Belohnung von außen zu bekommen.

Dies zeigt nach [**?**], dass das eigenständige Lernen von Fähigkeiten einen effektiven Mechanismus bereit stellt, um bei Herausforderungen beim Erkunden und mit geringem externen Reward Lernerfolge zu erzielen.

# 5 Curiosity und Diversity im Vergleich

# 6 Verwandte Arbeiten

# 7 Schluss

## Autorenschaft

David Jonathan Müller hat die Abschnitte **??**, **??** und **??** verfasst. Lukas Johannes Rieger hat die Abschnitte **??** und **??** verfasst. Den Abschnitt **??** haben beide Autoren gemeinsam verfasst. (In einer wissenschaftlichen Arbeit ohne Prüfungszweck stünde hier eine Danksagung!)