Reinforcement Learning

Grundlagen Grundbegriffe Grundübungen

Alfatraining Kurs 2026-01 Dozent: Karsten Flügge

Themen des Tages

Tag 15

Dreamer (sota 2025)

Novelty, curiosity, surprise: Exploration

RL *in* large language models *in* RL

Attention / Relational / Graph Neural Networks in RL

Pause

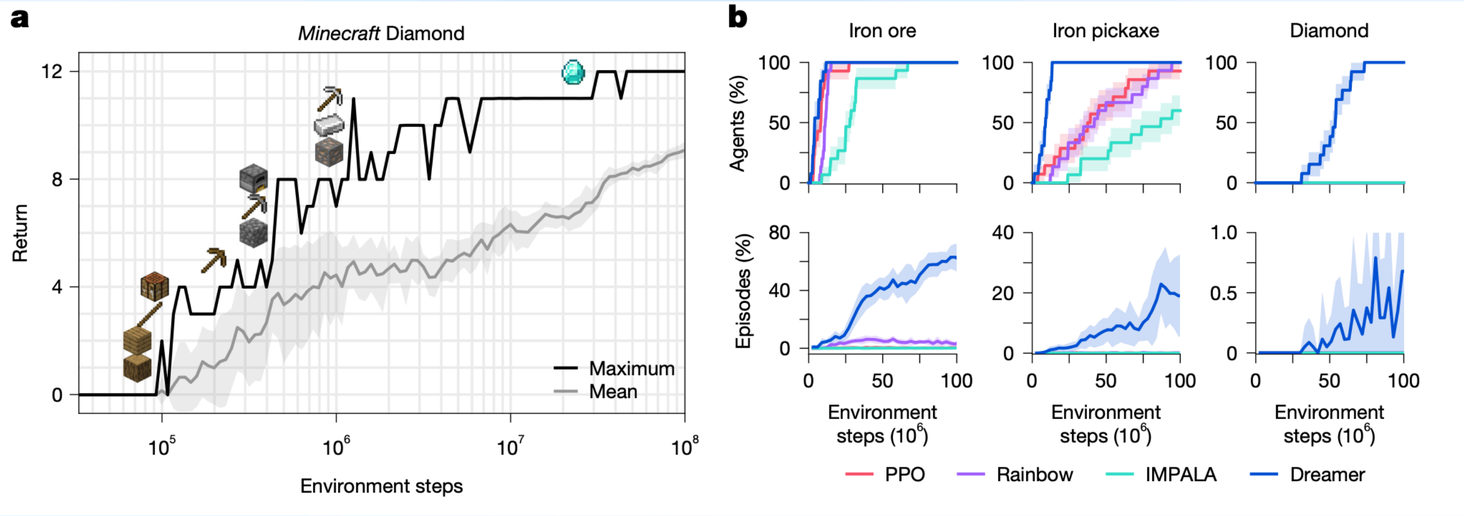
Dreamer v3

General algorithm that learns to **solve tasks** across a **wide range** of applications
outperforms specialized methods across over **150 diverse tasks,**
with a **single configuration:** no hyperparameter changes!

model of the environment improves its behaviour by **imagining** future scenario

Robustness techniques based on **normalization, balancing and transformations:**

- **EMA Exponential Moving Average …**
- **percentile clip**
- **return normalization** with a denominator limit
- **symlog squared error**
- **two-hot encoding** to approximate continuous scalars



Dreamer Komponenten

Sequence model:	$h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1})$
Encoder:	$z_t \sim q_\phi(z_t \mid h_t, x_t)$
Dynamics predictor:	$\hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t)$
Reward predictor:	$\hat{r}_t \sim p_\phi(\hat{r}_t \mid h_t, z_t)$
Continue predictor:	$\hat{c}_t \sim p_\phi(\hat{c}_t \mid h_t, z_t)$
Decoder:	$\hat{x}_t \sim p_\phi(\hat{x}_t \mid h_t, z_t)$

h Hidden state **h** (wie meta **rnn**?) ≈ **action**?
f (wie meta rnn?)
z latent state : encoded input state(?) x (context dependent like in Attention)
p world model "dynamics" p(z|h)=**z,r** predict state+reward
c continue/done flag?
x state back from latent z (f^{-1}=p??)

Dreamer

Sequence model:	$h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1})$
Encoder:	$z_t \sim q_\phi(z_t \mid h_t, x_t)$
Dynamics predictor:	$\hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t)$
Reward predictor:	$\hat{r}_t \sim p_\phi(\hat{r}_t \mid h_t, z_t)$
Continue predictor:	$\hat{c}_t \sim p_\phi(\hat{c}_t \mid h_t, z_t)$
Decoder:	$\hat{x}_t \sim p_\phi(\hat{x}_t \mid h_t, z_t)$

$$\mathcal{L}_{\text{pred}}(\phi) \doteq -\log p_\phi(x_t \mid z_t, h_t) - \log p_\phi(r_t \mid z_t, h_t) - \log p_\phi(c_t \mid z_t, h_t)$$

$$\mathcal{L}_{\text{dyn}}(\phi) \doteq \max(1, \mathrm{KL}[\mathrm{sg}(q_\phi(z_t \mid h_t, x_t)) \| p_\phi(z_t \mid h_t)])$$

$$\mathcal{L}_{\text{rep}}(\phi) \doteq \max(1, \mathrm{KL}[q_\phi(z_t \mid h_t, x_t) \| \mathrm{sg}(p_\phi(z_t \mid h_t))])$$

$$\mathcal{L}(\phi) \doteq E_{q_\phi}\left[\sum_{t=1}^{T} (\beta_{\text{pred}}\mathcal{L}_{\text{pred}}(\phi) + \beta_{\text{dyn}}\mathcal{L}_{\text{dyn}}(\phi) + \beta_{\text{rep}}\mathcal{L}_{\text{rep}}(\phi))\right],$$

Dreamer

$$\mathcal{L}_{\text{pred}}(\phi) \doteq -\log p_\phi(x_t|z_t,h_t) - \log p_\phi(r_t|z_t,h_t) - \log p_\phi(c_t|z_t,h_t)$$
$$\mathcal{L}_{\text{dyn}}(\phi) \doteq \max(1, \mathrm{KL}[\mathrm{sg}(q_\phi(z_t|h_t,x_t)) \| p_\phi(z_t|h_t)])$$
$$\mathcal{L}_{\text{rep}}(\phi) \doteq \max(1, \mathrm{KL}[q_\phi(z_t|h_t,x_t) \| \mathrm{sg}(p_\phi(z_t|h_t))])$$

KL Kullback-Leibler-Divergenz misst wie nah unsere neue policy(?) an der alten ist (so wie **PPO** mit log, mit clipping 1 ≈ 1.44 bits?)

Je kleiner die KL-Divergenz, desto ähnlicher sind die Policies:

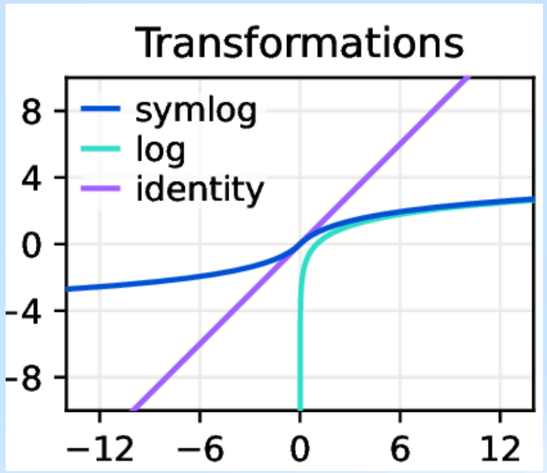
$$D_{\mathrm{KL}}(\pi_{\text{alt}} \parallel \pi_{\text{neu}}) = \sum_a \pi_{\text{alt}}(a \mid s) \log \left(\frac{\pi_{\text{alt}}(a \mid s)}{\pi_{\text{neu}}(a \mid s)} \right)$$

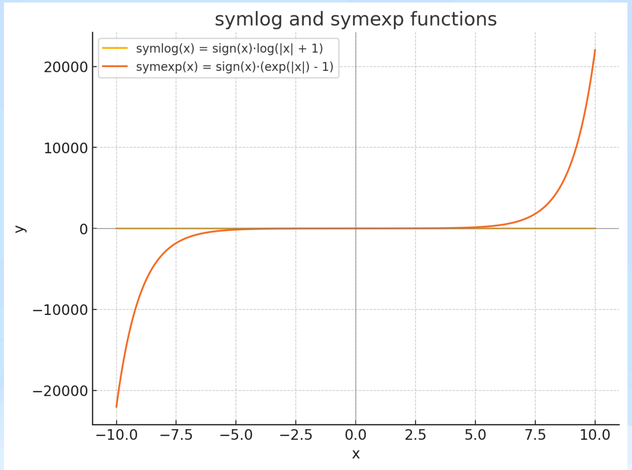
This disables them while they are already minimized well to focus learning on the prediction loss

Dreamer

$$\mathrm{symlog}(x) \doteq \mathrm{sign}(x)\log(|x|+1)$$
$$\mathrm{symexp}(x) \doteq \mathrm{sign}(x)(\exp(|x|)-1)$$

"leaky tanh"





symlog(x) = sign(x) · log(|x| + 1) **wächst langsam** für große |x| (ähnlich wie log) **symmetrisch**
symexp(x) = sign(x) · (exp(|x|) – 1) **wächst sehr schnell** für große |x| (ähnlich wie exp) **symmetrisch**

Dreamer

Zwei **Twohot encodings:**
1.) Zwei Einsen erlaubt
2.) Zwei Werte summieren sich zu 1!
Twohot erlaubt darstellen von **kontinuierlichen Aktionen,** indem man zwischen zwei outputs interpoliert
Wahrscheinlichkeiten nicht-binär (aber **summierend zu 1**)

Die zwei Ausschlag gebenden Aktivierungen werden behalten

$$\text{twohot}(x)_i \doteq \begin{cases} |b_{k+1}-x|/|b_{k+1}-b_k| & \text{if } i=k \\ |b_k-x|/|b_{k+1}-b_k| & \text{if } i=k+1 \\ 0 & \text{else} \end{cases} \qquad k \doteq \sum_{j=1}^{|B|} \delta(b_j < x)$$

[f b l r] Aktivierung = Mittelung aus zwei größten Softmax Werten

[f b l r] = [0 0 .8 .2] = fahre etwas nach links

v = binominial(f,b)

f=.5 b=.5 v = 0

f=1 b=0 v = 1

f=.8 b=.2 v = .6

Dreamer

sg "stop gradient"

In PyTorch: `x.detach()`

In TensorFlow: `tf.stop_gradient(x)`

In JAX: `jax.lax.stop_gradient(x)`

Kontrollierter Gradienten Flow.

$$\mathcal{L}_{\text{pred}}(\phi) \doteq -\log p_\phi(x_t|z_t,h_t) - \log p_\phi(r_t|z_t,h_t) - \log p_\phi(c_t|z_t,h_t)$$

$$\mathcal{L}_{\text{dyn}}(\phi) \doteq \max(1, \text{KL}[\text{sg}(q_\phi(z_t|h_t,x_t)) \| p_\phi(z_t|h_t)])$$

$$\mathcal{L}_{\text{rep}}(\phi) \doteq \max(1, \text{KL}[q_\phi(z_t|h_t,x_t) \| \text{sg}(p_\phi(z_t|h_t))])$$

Dreamer

Regularisierung:

EMA Exponential Moving Average

- Maintain a **target actor** with EMA updates during training:

$$\theta_{\text{EMA}} \leftarrow \tau \cdot \theta_{\text{EMA}} + (1 - \tau) \cdot \theta$$

with $\tau \in [0.99, 0.999]$

<> Elastic Net?

Dreamer v3

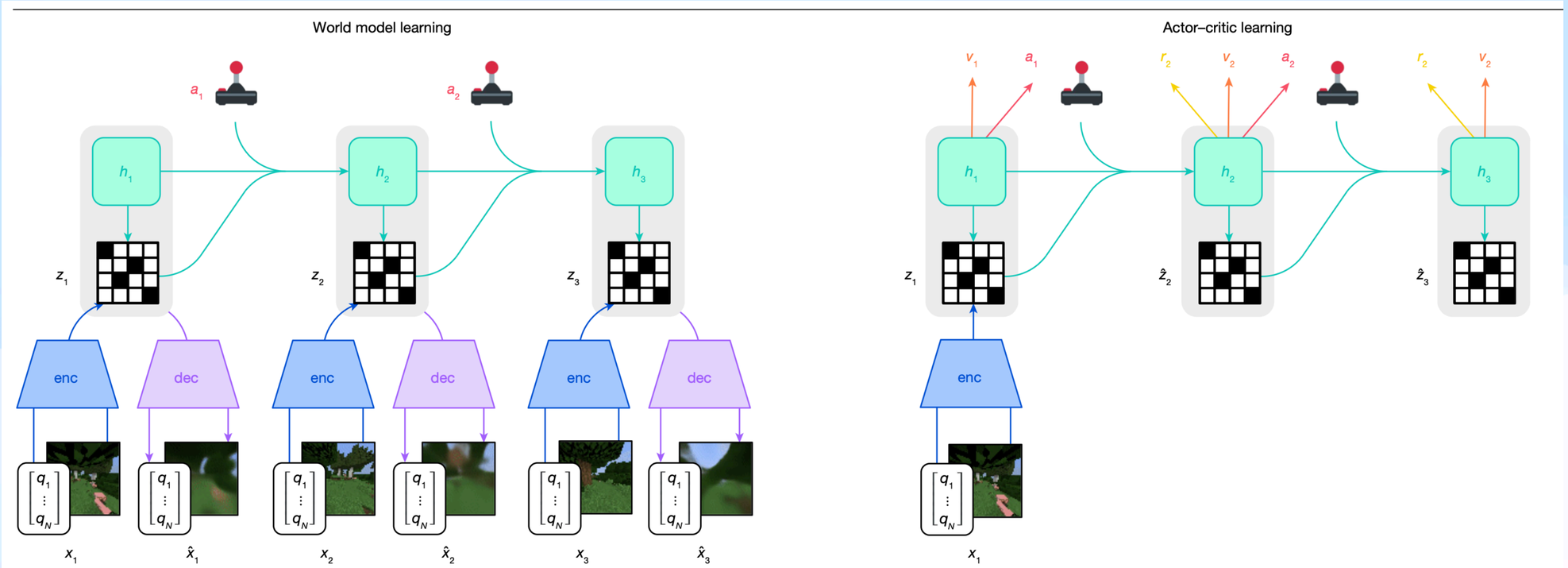


Fig. 1 | Training process of Dreamer. The world model encodes sensory inputs x_t using the encoder (enc) into discrete representations z_t that are predicted by a sequence model with recurrent state h_t given actions a_t. The inputs are reconstructed as $\hat{x}_t$ using the decoder (dec) to shape the representations. The actor and critic predict actions a_t and values v_t and learn from trajectories of abstract representations $\hat{z}_t$ and rewards r_t predicted by the world model.

Baselines

PPO
IMPALA and Rainbow for Minecraft MineRL v0.4.4 abstract crafting actions (cheat?)

To investigate whether Dreamer can scale robustly, we train **6 model sizes** ranging from **12 million to 400 million** parameters, as well as different replay ratios.

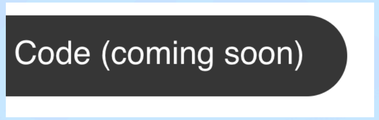
Trainable on **one GPU** in **10 days.**

Pause

Reinforcement Learning with Foundation Priors:
Let the Embodied Agent Efficiently Learn on Its Own

https://arxiv.org/abs/2310.02635

https://yewr.github.io/rlfp/



Dumb exploration

Bisherige Exploration war **zufällig** und **ungerichtet**:

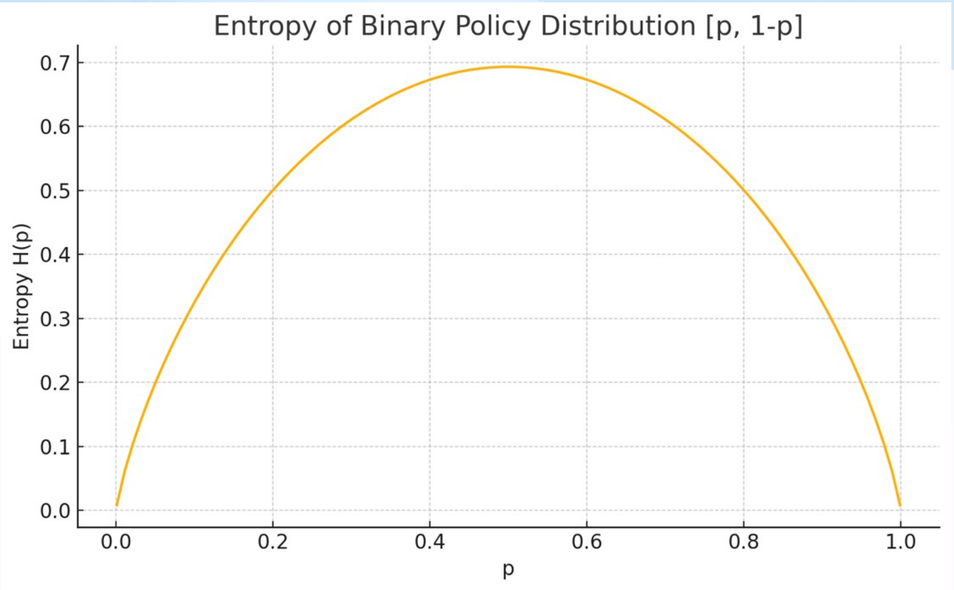
ε-greedy zufällig per Definition

…

Entropie:

$$\mathcal{L}_{\text{actor}} = \mathbb{E}[A(s,a) + \alpha \cdot \mathcal{H}(\pi(\cdot|s))]$$

- $\mathcal{H}(\pi)$: Entropy of the policy, i.e. $-\sum \pi(a|s) \log \pi(a|s)$
- Promotes **action diversity** (randomness), especially early in training



Does **not modify rewards**, but affects the gradient

- **No environment signal** involved
- **No model** involved
- **No reward** shaping

Novelty, surprise: curiosity exploration

- **Novelty** (e.g., **count-based** or hash-based methods)
- Curiosity via **surprise** (e.g., **prediction error** of *model*s) <>
- Curiosity via **uncertainty** (e.g., Bayesian uncertainty or **ensemble disagreement**)

Novelty, surprise: curiosity exploration

- **Novelty** (e.g., **count-based** or hash-based methods)

  zähle wie oft ein State erreicht wurde!

  diskret oder über continuous neighborhood **estimate, distance**

- Curiosity via **surprise** (e.g., **prediction error** of forward models) <>
- Curiosity via **uncertainty** (e.g., Bayesian uncertainty or **ensemble disagreement**)

sigmoid peeks vs near uniform

Steuerung zum Beispiel in **policy direkt** oder über **reward**

$$r_t^{\text{aug}} = r_t^{\text{env}} + \beta \cdot r_t^{\text{explore}}$$

<> **Advantage** orthogonal objectives:

advantage optimizes **known rewards**,

exploration bonuses **bias** toward acquiring new knowledge

Novelty, surprise: curiosity exploration

- Curiosity via **surprise** (e.g., **prediction error** of forward models) <>

Steuerung zum Beispiel in **policy direkt** oder über **reward** $r_t^{\text{aug}} = r_t^{\text{env}} + \beta \cdot r_t^{\text{explore}}$

Führt so eine Belohnung dazu dass absichtlich falsche Vorhersagen getätigt werden??

Novelty, surprise: curiosity exploration

• Curiosity via **surprise** (e.g., **prediction error** of forward models) <>

Steuerung zum Beispiel in **policy direkt** oder über **reward**

Führt so eine Belohnung dazu dass absichtlich falsche Vorhersagen getätigt werden?

JA! Problematisch:

• **Exploration wird Selbstzweck**

→ Agent sucht maximales Unwissen statt sinnvolle Information

• **Loophole bei reward shaping**

→ Agent trickst das Lernsystem aus, um hohe intrinsische Belohnung zu erhalten

• **Non-stationäres Ziel**

→ Belohnung basiert auf einem sich ständig verändernden Modell

Gegenmaßnahmen:

• **Im Latent space arbeiten**

• **Bonus nur temporär** oder nur in **frühen Phasen**

• **Decay des Intrinsic-Reward-Gewichts**

• **Maskierung chaotischer Zustände** (z. B. durch epistemische Unterscheidung)

• **Use surprise only when learning is possible** (→ predictability prior)

Belohne nur Zustände mit **hoher epistemischer Unsicherheit** (nicht bloß hoher Fehler)

Bild 8.5 Das Problem des verrauschten Fernsehers ist ein theoretisches und praktisches Problem, bei dem ein Reinforcement-Learning-Agent mit einem einfachen Sinn für Neugierde von einem verrauschten Fernseher in den Bann gezogen wird und ihn ewig anstarrt. Das liegt daran, dass er von Natur aus durch Unvorhersehbarkeit belohnt wird, und weißes Rauschen sehr unberechenbar ist.

Novelty, curiosity, surprise: Exploration

- **Novelty** (e.g., **count-based** or hash-based methods)
- **Curiosity** (e.g., **prediction error** of forward models) <>
- **Surprise** (e.g., Bayesian uncertainty or **ensemble disagreement**)

NICHT **explizit** in Dreamer! Aber **implizit:**

Dreamer relies purely on:

1. **Stochastic world models** (e.g., RSSM)

→ Intrinsic variability in latent imagination leads to **structured exploration**.

2. **Entropy regularization** in the actor loss:

$$\mathbb{E}_{a\sim\pi}[\log \pi(a|s)]$$

→ Encourages **policy entropy**, indirectly promoting exploration.

Dreamer avoids these by **not adding any intrinsic reward**

- Leverages **learned latent dynamics** for **long-horizon planning**

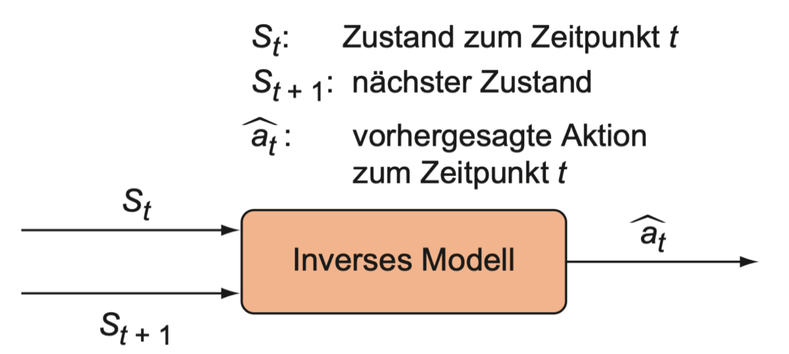
„Curiosity-driven Exploration by Self-supervised Prediction“ (2017)

einer der am einfachsten zu implementierenden Algorithmen

Zusätzlich zur Dynamik, also zum Modell

p(s,a)=>s' bauen wir die Rückrichtung

f(s,s')=>a welche versucht vorherzusagen mithilfe von **welcher Aktion zum nächsten Zustand** gefunden wurde.



Wir wollen uns nur auf '**wichtige Aspekte**' des Zustandes konzentrieren, daher

Encoder s=>**z**

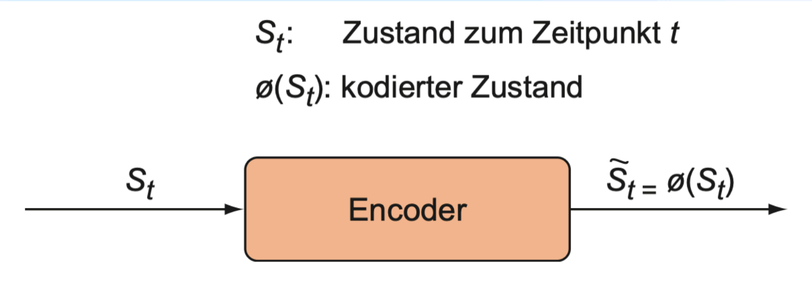


Bild 8.5 Das Problem des verrauschten Fernsehers ist ein theoretisches und praktisches Problem, bei dem ein Reinforcement-Learning-Agent mit einem einfachen Sinn für Neugierde von einem verrauschten Fernseher in den Bann gezogen wird und ihn ewig anstarrt. Das liegt daran, dass er von Natur aus durch Unvorhersehbarkeit belohnt wird, und weißes Rauschen sehr unberechenbar ist.

„Curiosity-driven Exploration by Self-supervised Prediction“ (2017)

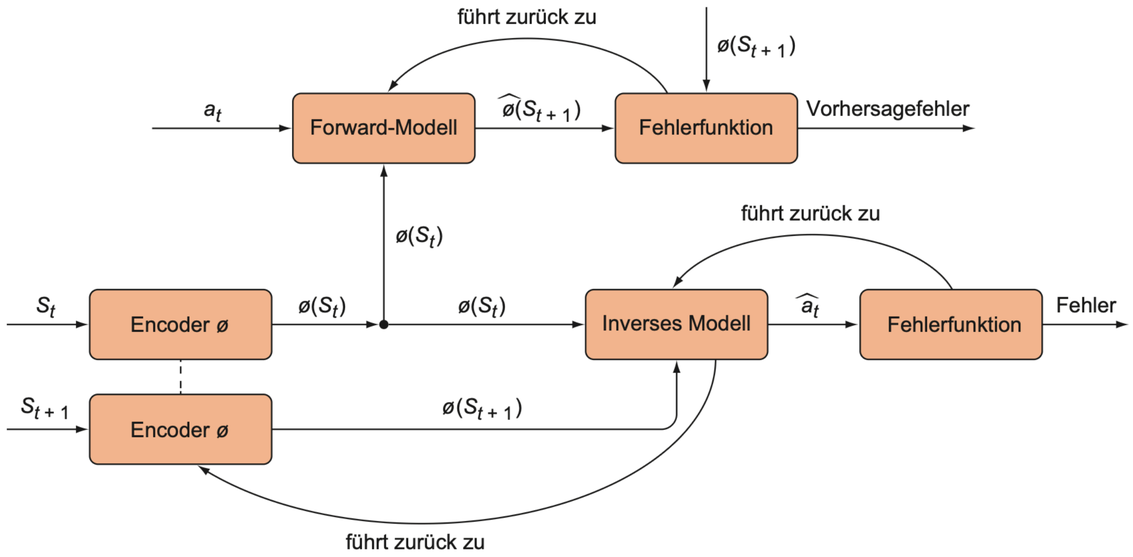


Bild 8.10 Das Neugierde-Modul. Zuerst kodiert der Encoder die Zustände S_t und S_{t+1} in niedrigdimensionale Vektoren, $\phi(S_t)$ bzw. $\phi(S_{t+1})$. Diese kodierten Zustände werden an das Forward- und das inverse Modell weitergegeben. Beachten Sie, dass das inverse Modell auf das Encoder-Modell zurückgeführt wird und es somit durch seinen eigenen Fehler trainiert. Das Forward-Modell wird durch Backpropagieren aus seiner eigenen Fehlerfunktion trainiert, aber es wird nicht wie das inverse Modell zum Encoder zurückgeführt. Dadurch wird sichergestellt, dass der Encoder lernt, Zustandsdarstellungen zu erzeugen, die nur für die Vorhersage, welche Aktion durchgeführt wurde, nützlich sind. Der schwarze Kreis zeigt einen Kopiervorgang an, bei dem die Ausgabe vom Encoder kopiert und die Kopien an das Forward- und das inverse Modell weitergegeben werden.

Pause

Later work (e.g., **Plan2Explore**, **DreamerV3 with pretraining**) does explore **intrinsic reward extensions** — but not Dreamer V1/V2 in their core forms.

Contrasted with methods that do use explicit signals:

Method	Exploration Signal
ICM (Curiosity, 2017)	Prediction error of forward model
RND (Burda et al., 2018)	Disagreement with fixed random net
Count-based (MBIE-EB)	State visitation counts
Pseudo-counts	Density models (PixelCNN, etc.)
Disagreement (e.g. PETS)	Variance in model predictions

Dreamer avoids these by **not adding any intrinsic reward**:

$$r_t^{\text{total}} = r_t^{\text{env}} \quad \text{(no exploration bonus)}$$

Pause