Questo grafico mostra, episodio dopo episodio, il reward “grezzo” che il tuo agente REINFORCE ottiene nell’ambiente MuJoCo Hopper. Ci sono tre aspetti principali da notare:

**\*\*\*Varianza molto alta\*\*\***

Ogni punto è il return totale di un singolo episodio (senza alcuna media o smoothing), e quindi vedi oscillazioni enormi: alcuni episodi il robot salta davvero bene (picchi fino a 400–500), in altri addirittura cade subito (reward vicino a 0).

Questa è la caratteristica principale di REINFORCE puro: il gradiente di politica stimato via Monte Carlo ha varianza altissima perché dipende da una singola traiettoria. **VEDI DOPO**

**\*\*\*Lento miglioramento del valore atteso\*\*\***

Nelle prime decine di migliaia di episodi (fino a ≃30 k) il comportamento è sostanzialmente casuale e i reward medi si assestano tra 50 e 150.

Dopo ≃30 k–40 k episodi inizia a emergere un trend di crescita: i reward oscillano su valori più alti (200–300), segno che la politica ha imparato qualche sequenza utile di mosse.

Intorno a 60 k episodi si vede un salto netto verso reward di 350–400 in alcuni episodi, ma non c’è ancora una piena stabilità.

**\*\*\*Instabilità anche a lungo termine\*\*\***

Anche dopo 80 k–90 k episodi, dove a volte si toccano valori di reward superiori a 500, la curva scende bruscamente in altri episodi.

Questo indica che la policy non è “stabile”: basta una piccola modifica dei parametri (o un episodio particolarmente sfortunato) per farla tornare a comportamenti pessimi.

**Possibili rimedi per un apprendimento più stabile**

Baseline / Critic: aggiungere un termine di baseline (ad es. un valore stimato dallo stesso network) riduce la varianza del gradiente.

Smoothed curve: tracciare la media mobile (es. su 100 episodi) ti darebbe una visione più chiara del progresso “medio” e non solo dei singoli outlier.

Algorithm tweaks: passare a metodi actor-critic (A2C, PPO, ecc.) spesso porta a convergenza più veloce e stabile grazie a stime del valore più accurate e aggiornamenti più conservativi.

In sintesi, il grafico conferma sia la capacità di REINFORCE di imparare—visto il trend crescente dei reward massimi—sia la sua grande difficoltà a produrre una policy affidabile a causa dell’elevata varianza degli aggiornamenti.

**SPIEGAZIONE DI PRIMA**

Quando usi REINFORCE “puro” (senza baseline né altri accorgimenti), ogni aggiornamento dei pesi della policy si basa su un’unica simulazione completa. Il gradiente stimato è:

dove il return totale a partire dal passo ttt è:

**Perché questo genera alta varianza?**

1. **Dipendenza dalle ricompense future casuali**  
   Il valore GtG\_{t}Gt​ ingloba tutte le ricompense fino alla fine dell’episodio; variazioni piccole negli step iniziali si amplificano lungo l’orizzonte.
2. **Prodotto di due termini aleatori**  
   Anche il termine ∇θlog⁡πθ(at∣st)\nabla\_{\theta}\log\pi\_{\theta}(a\_{t}\mid s\_{t})∇θ​logπθ​(at​∣st​) è casuale. L’aggiornamento è il prodotto di due variabili aleatorie, amplificando la dispersione del gradiente.
3. **Orizzonte lungo**  
   Più è grande TTT, più rumore si accumula nella somma.

**Conseguenze pratiche**

* **Oscillazioni estreme**: un episodio può dare reward vicino a zero, il successivo oltre 500.
* **Lentezza di convergenza**: per stabilizzare lo stimatore servirebbero molti campioni, aumentando il costo computazionale.

**Come ridurre la varianza**

1. **Baseline / Critic**  
   Sottrarre da GtG\_{t}Gt​ una baseline b(st)b(s\_{t})b(st​) non introduce bias, ma riduce la varianza. L’aggiornamento diventa:
2. **Advantage**  
   Definiamo l’advantage come:

usando V(st)V(s\_{t})V(st​) stimato da una rete critica.

1. **Mini-batch di episodi**  
   Accumuli gradienti su un batch di NNN traiettorie e fai un unico update con la media:
2. **Algoritmi trust-region o clipping**  
   Metodi come PPO limitano la variazione di πθ\pi\_{\theta}πθ​ ad ogni passo, rendendo l’apprendimento più stabile.

In questo modo ottieni un gradiente meno rumoroso e una curva di reward mediamente più regolare.