Trust Region Policy Optimization (TRPO)

Shusen Wang

http://wangshusen.github.io/

Optimization Basics

Gradient Ascent

Problem: Find $\boldsymbol{\theta}^\star = \operatorname*{argmax}_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$.

Gradient Ascent

Problem: Find $\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \; J(\boldsymbol{\theta})$.

- **Gradient ascent repeats:**

  1. At $\boldsymbol{\theta}_{\text{old}}$, compute gradient $\mathbf{g} = \frac{\partial \, J(\boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}}$.
  2. Gradient ascent: $\boldsymbol{\theta}_{\text{new}} \leftarrow \boldsymbol{\theta}_{\text{old}} + \alpha \cdot \mathbf{g}$.

Stochastic Gradient Ascent

Problem: Find $\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ J(\boldsymbol{\theta})$.

- Assume $J(\boldsymbol{\theta}) = \mathbb{E}_S[V(S; \boldsymbol{\theta})]$.

Stochastic Gradient Ascent

Problem: Find $\boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \; J(\boldsymbol{\theta})$.

- Assume $J(\boldsymbol{\theta}) = \mathbb{E}_S[V(S; \boldsymbol{\theta})]$.
- **Stochastic gradient ascent repeats:**
  1. $s \leftarrow$ random sampling.
  2. At $\boldsymbol{\theta}_{\text{old}}$, compute gradient $\mathbf{g} = \frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}}$.
  3. Gradient ascent: $\boldsymbol{\theta}_{\text{new}} \leftarrow \boldsymbol{\theta}_{\text{old}} + \alpha \cdot \mathbf{g}$.

Trust Region

Problem: Find $\boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, J(\boldsymbol{\theta})$.

- Let $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$ be a neighborhood of $\boldsymbol{\theta}_{\text{old}}$

Trust Region

Problem: Find $\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \; J(\boldsymbol{\theta})$.

- Let $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$ be a neighborhood of $\boldsymbol{\theta}_{\text{old}}$, e.g.,

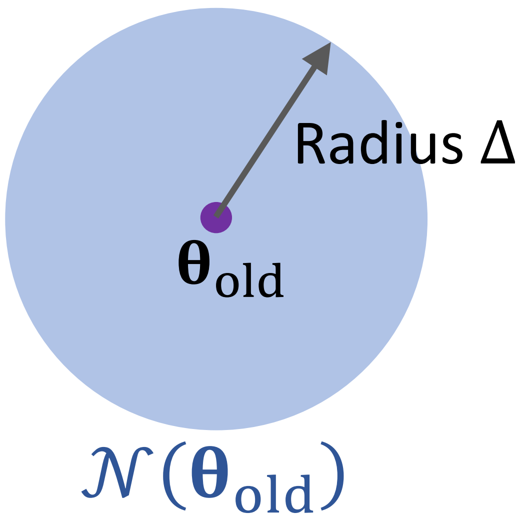
$$\mathcal{N}(\boldsymbol{\theta}_{\text{old}}) = \left\{ \boldsymbol{\theta} \;\middle|\; \|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{old}}\|_2 \leq \Delta \right\}.$$

Trust Region

Problem: Find $\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ J(\boldsymbol{\theta})$.

- Let $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$ be a neighborhood of $\boldsymbol{\theta}_{\text{old}}$, e.g.,

$$\mathcal{N}(\boldsymbol{\theta}_{\text{old}}) = \left\{ \boldsymbol{\theta} \,\middle|\, \|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{old}}\|_2 \leq \Delta \right\}.$$



Trust Region

Problem: Find $\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \; J(\boldsymbol{\theta})$.

- Let $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$ be a neighborhood of $\boldsymbol{\theta}_{\text{old}}$, e.g.,

$$\mathcal{N}(\boldsymbol{\theta}_{\text{old}}) = \left\{ \boldsymbol{\theta} \;\middle|\; \left\|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{old}}\right\|_2 \leq \Delta \right\}.$$

- If we have a function, $L(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}})$, that well approximates $J(\boldsymbol{\theta})$ in $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$, then $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$ is called "trust region".

Trust Region Algorithms

Problem: Find $\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ J(\boldsymbol{\theta})$.

Trust region algorithms

Trust Region Algorithms

Problem: Find $\boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ J(\boldsymbol{\theta})$.

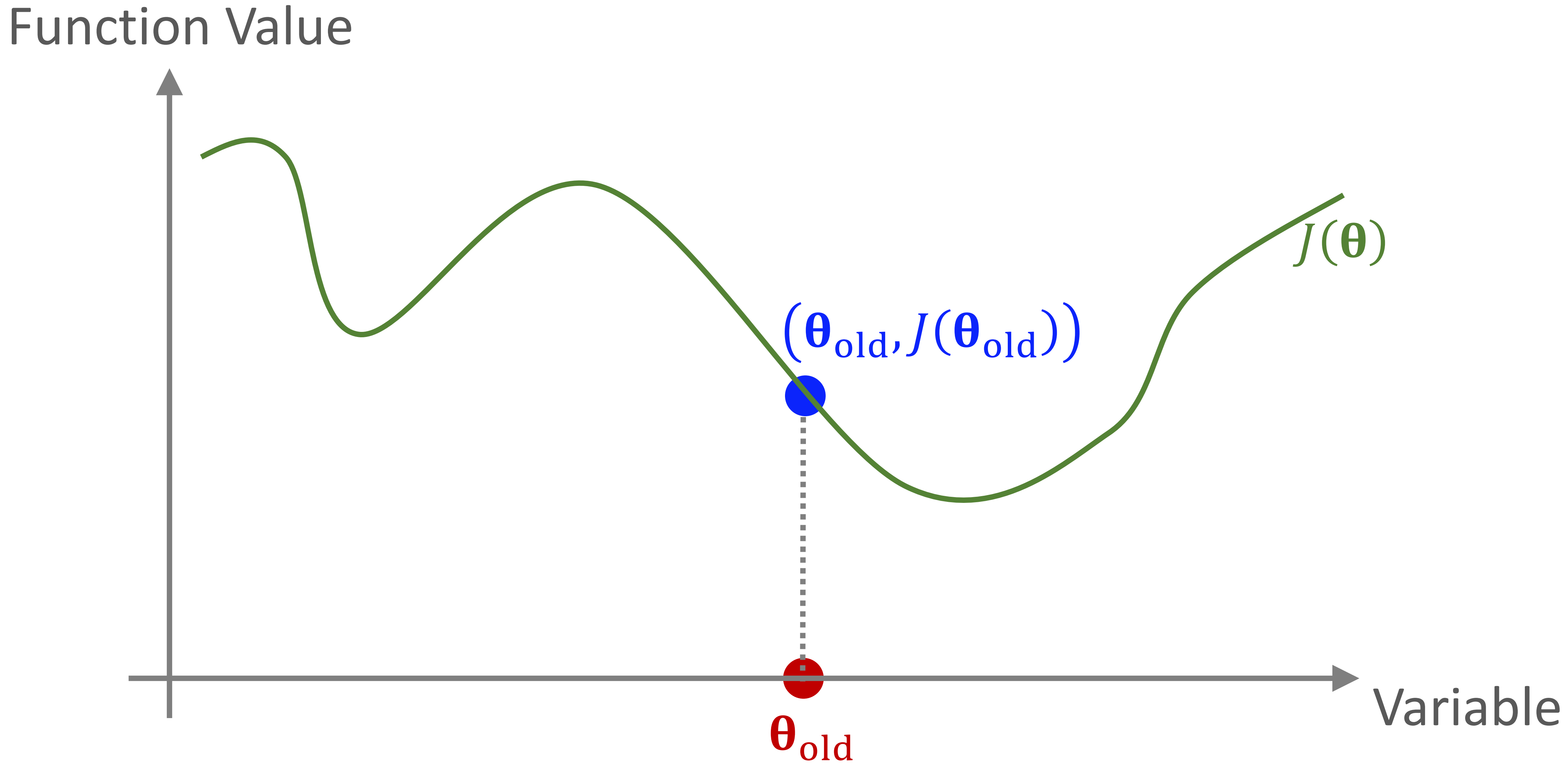
Trust region algorithms repeat:

1. **Approximation:** Given $\boldsymbol{\theta}_{\text{old}}$, construct $L(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}})$, which is an approximation to $J(\boldsymbol{\theta})$ in $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$.

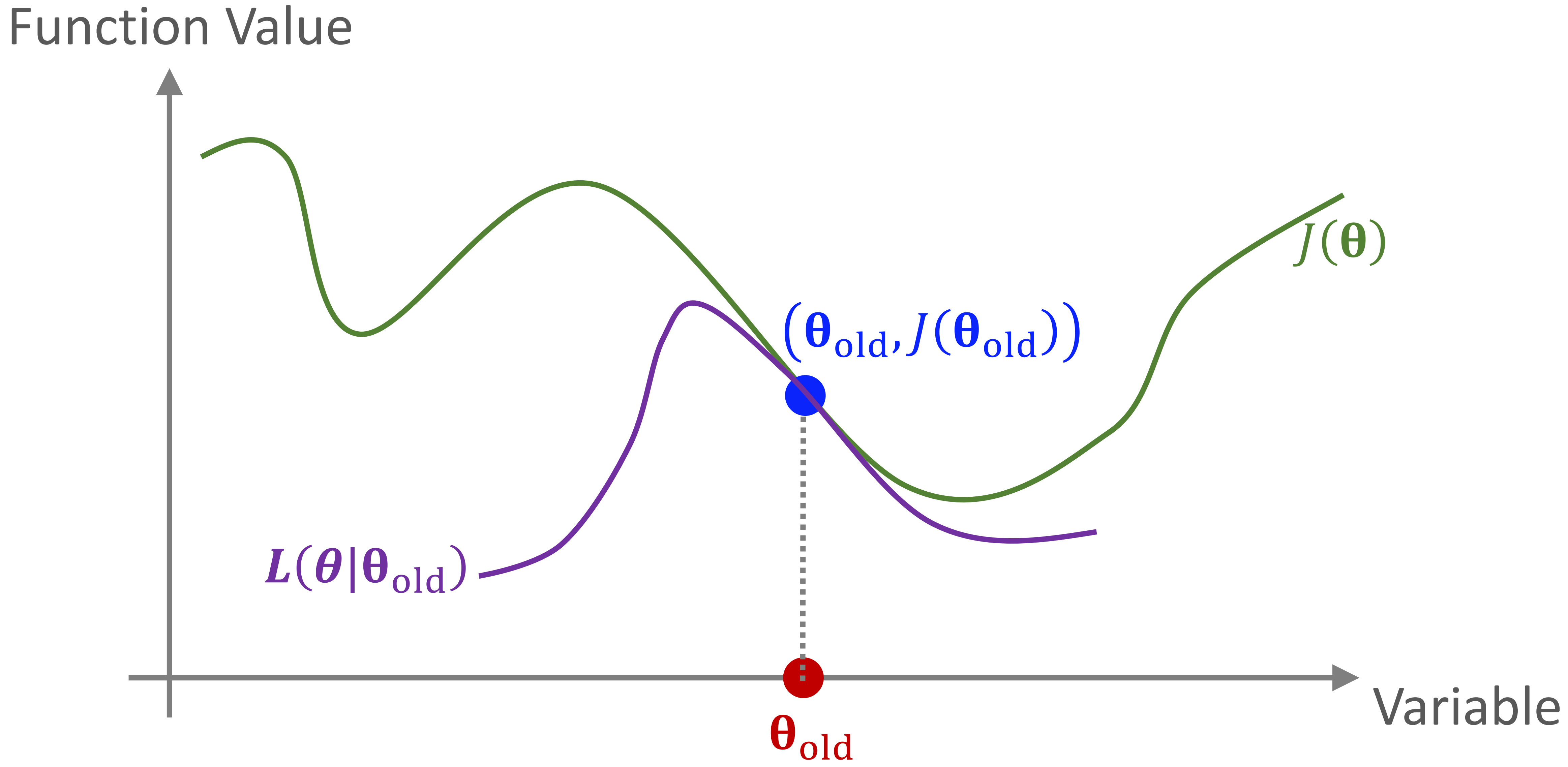
2. **Maximization:** In the trust region, find $\boldsymbol{\theta}_{\text{new}}$ by:

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \underset{\boldsymbol{\theta} \in \mathcal{N}(\boldsymbol{\theta}_{\text{old}})}{\operatorname{argmax}}\ L(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}).$$

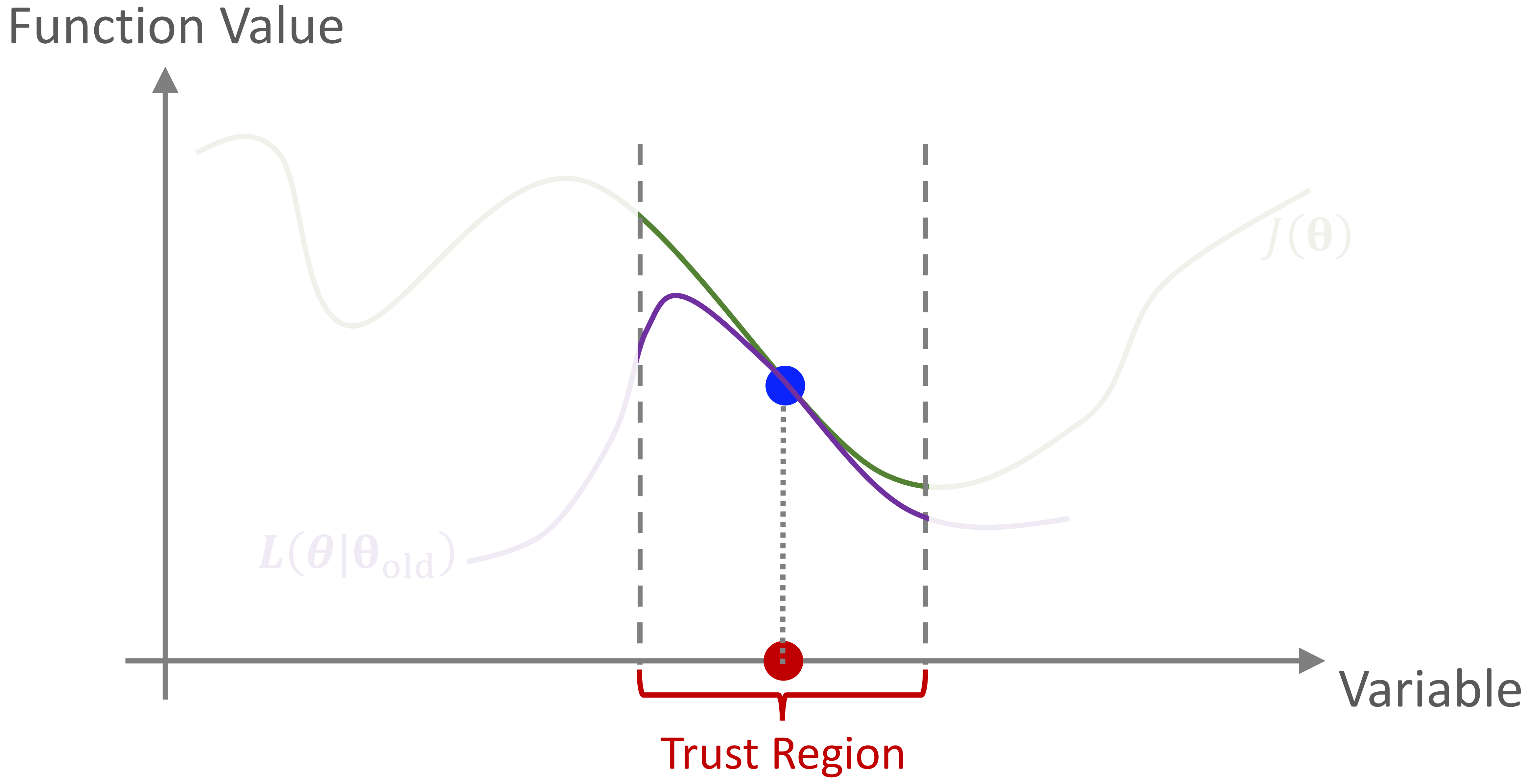
Trust Region Algorithms



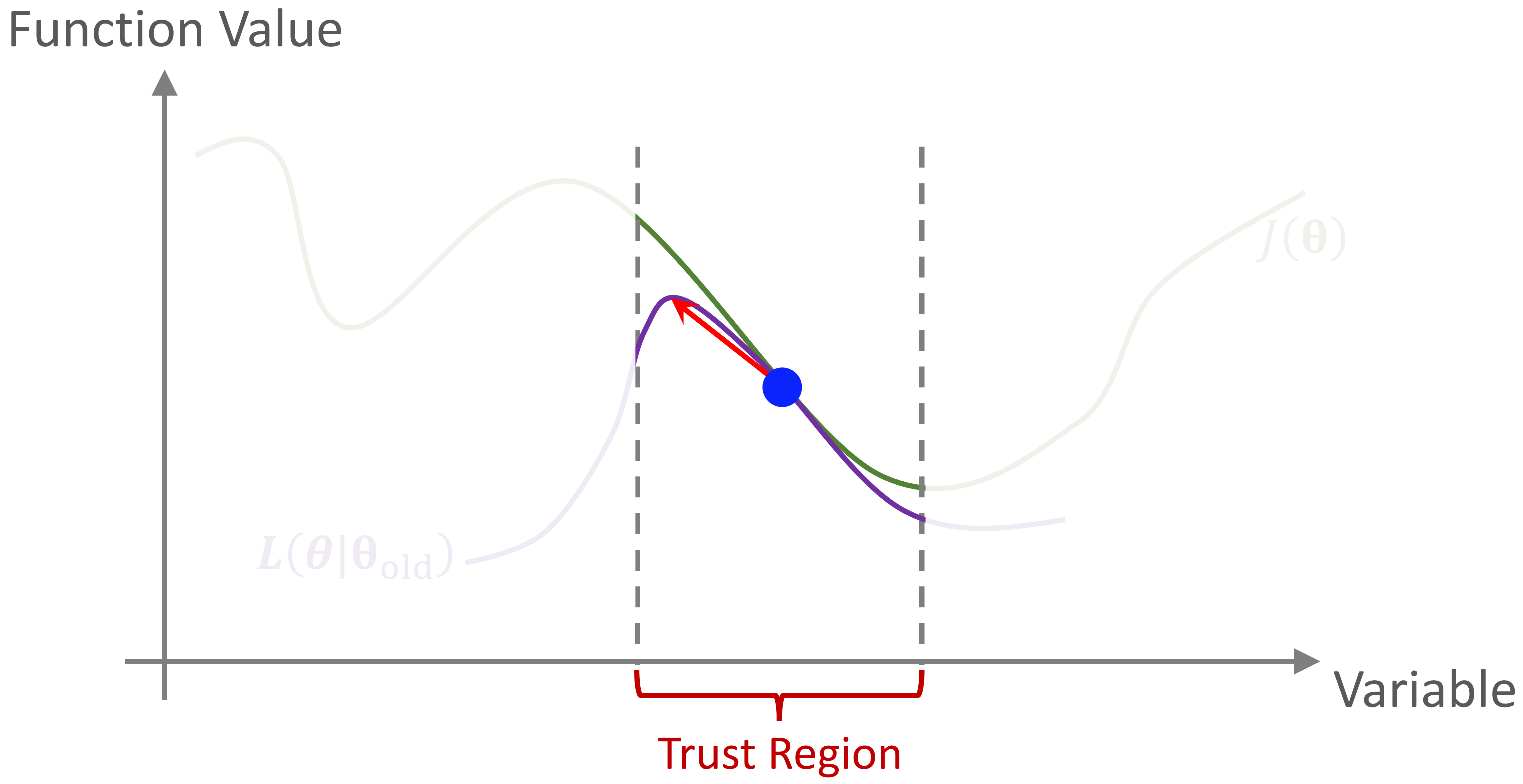
Trust Region Algorithms



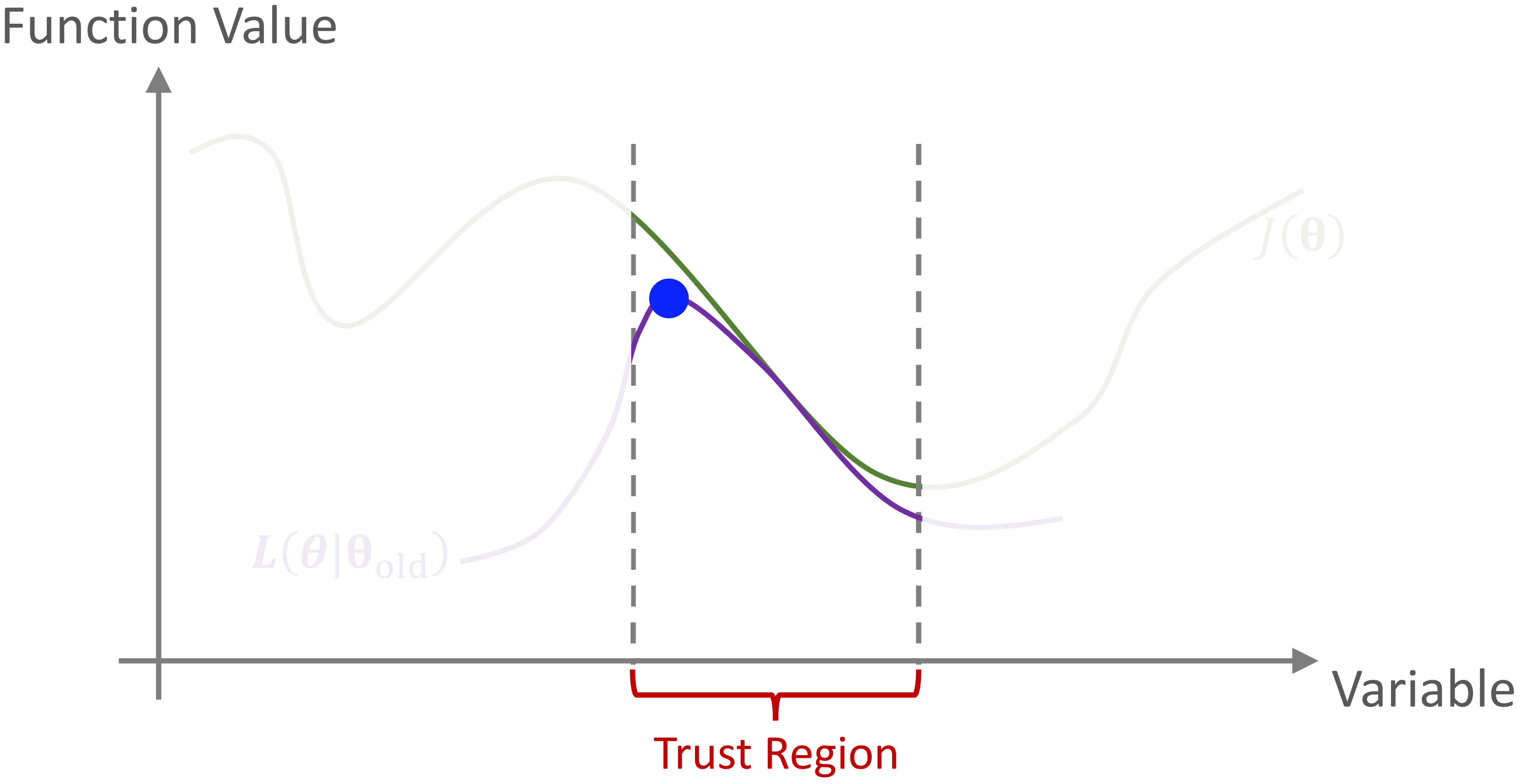
Trust Region Algorithms



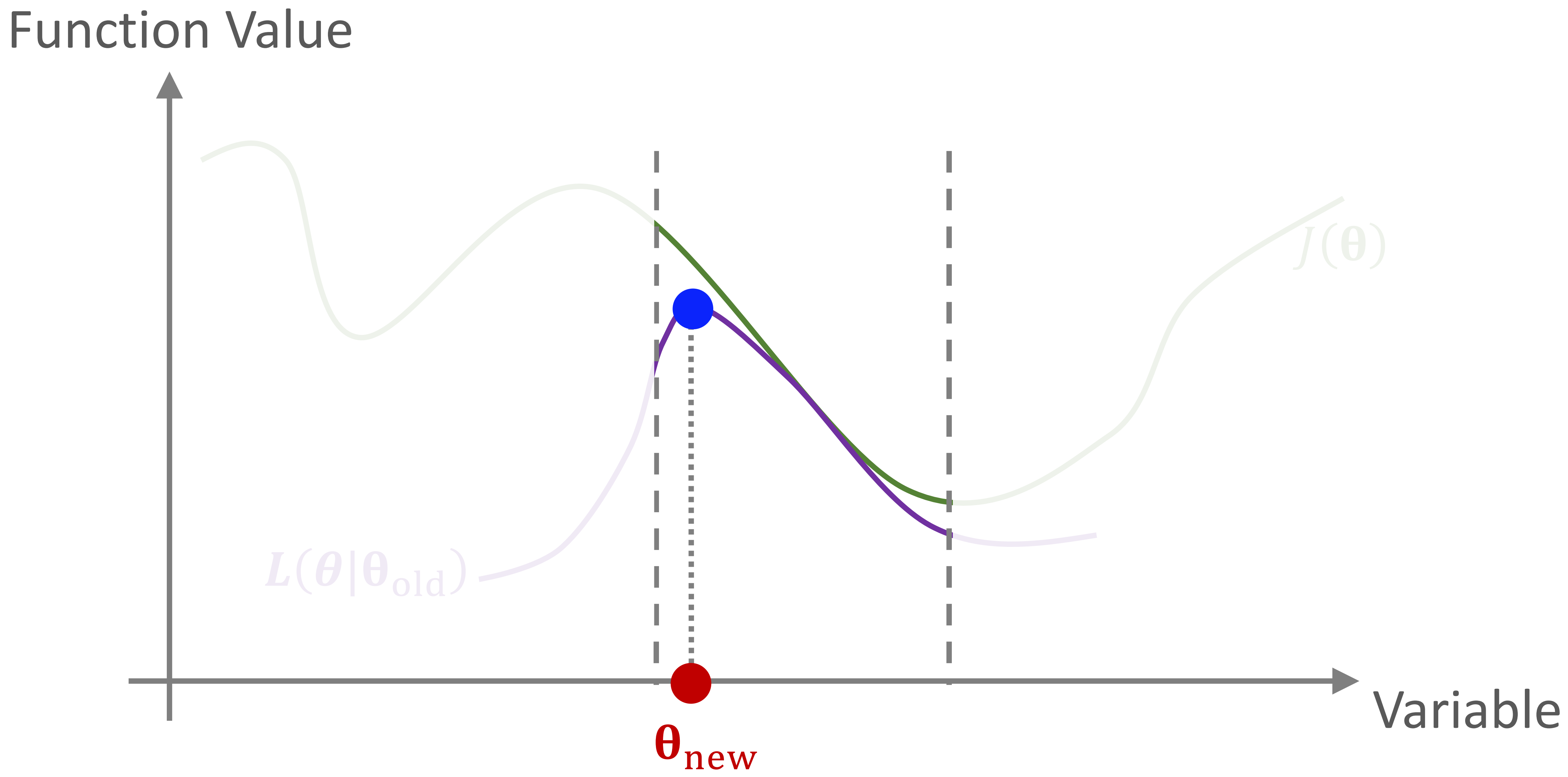
Trust Region Algorithms



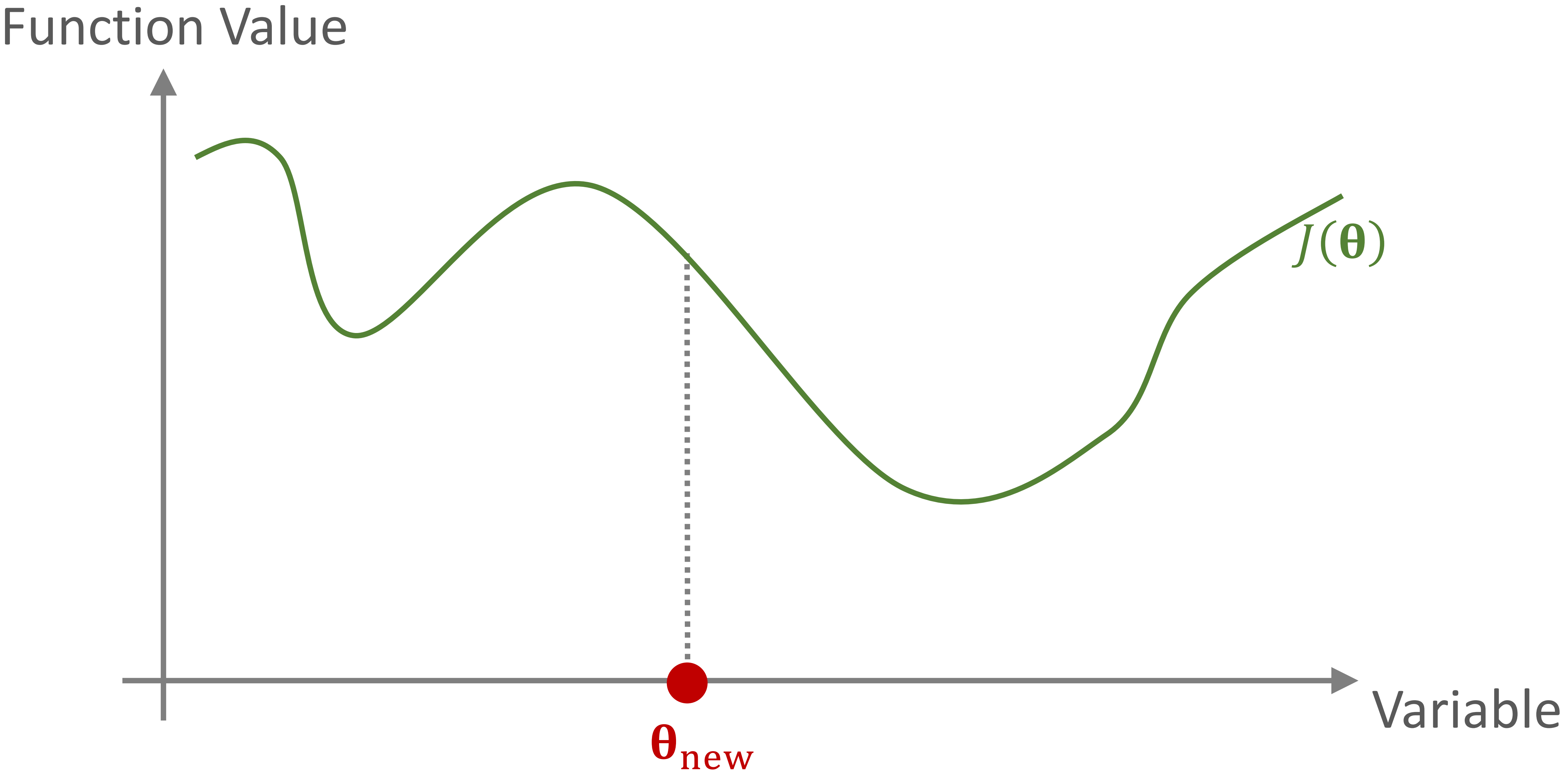
Trust Region Algorithms



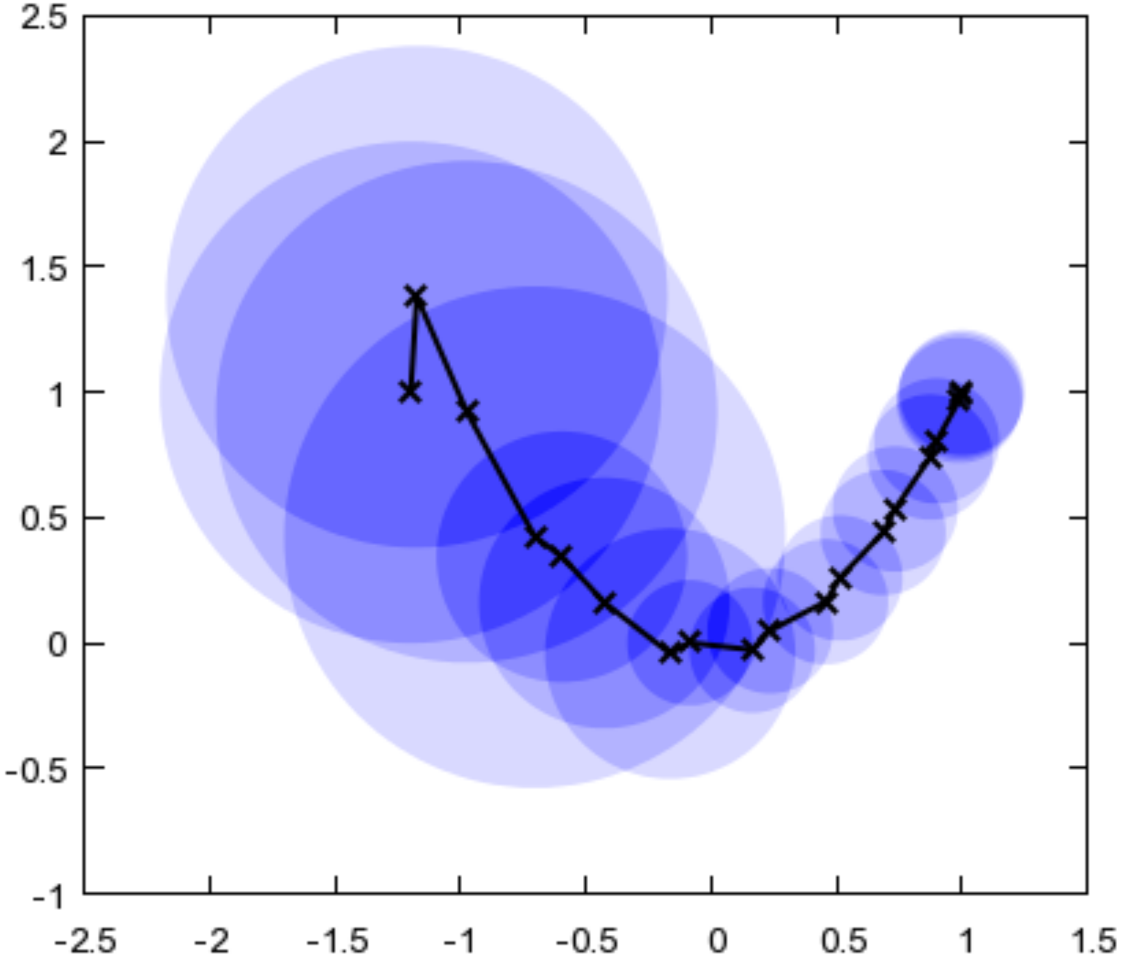
Trust Region Algorithms



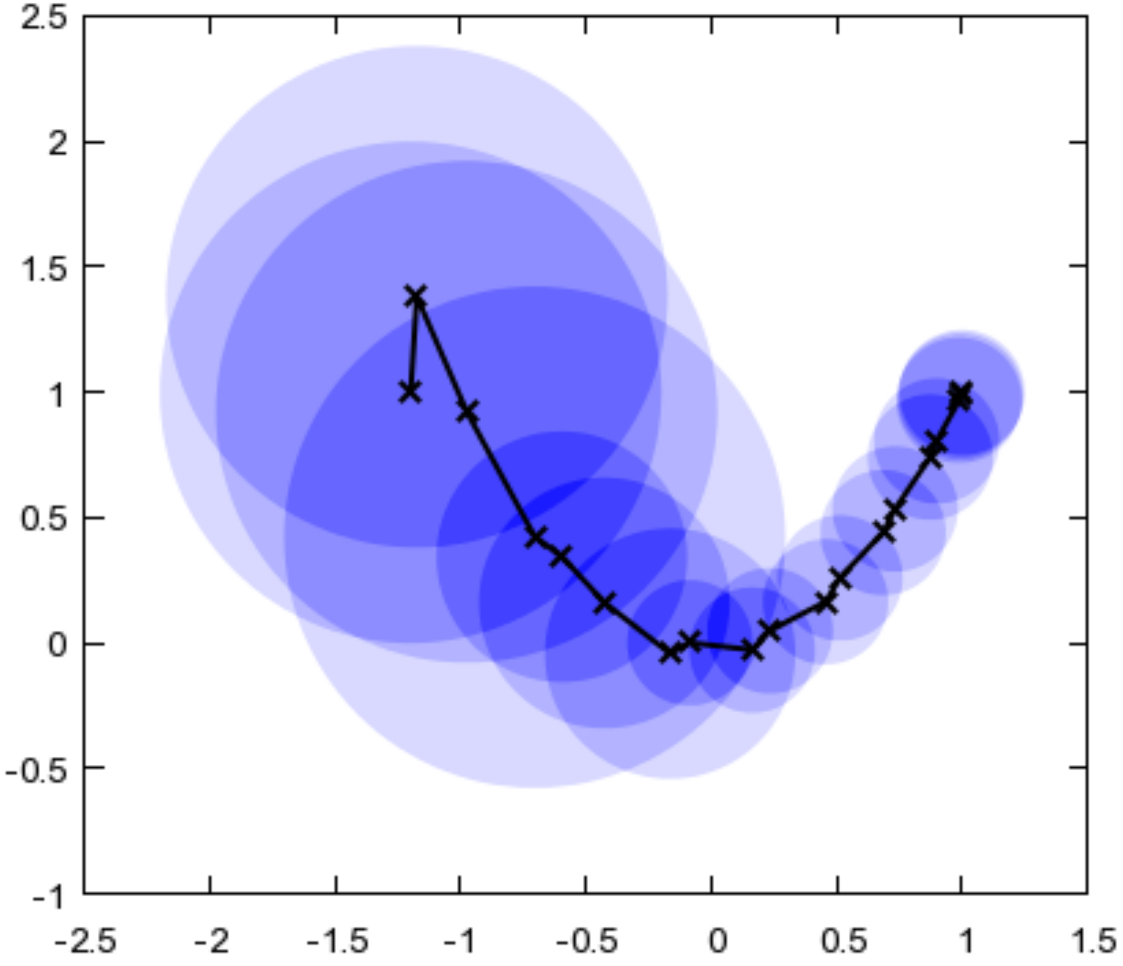
Trust Region Algorithms



Trust Region Algorithms



Trust Region Algorithms



Trust region algorithms repeat:

1. **Approximation:** Given $\boldsymbol{\theta}_{\text{old}}$, construct L which approximates J in the neighborhood of $\boldsymbol{\theta}_{\text{old}}$.

2. **Maximization:** In the trust region (i.e., the neighborhood of $\boldsymbol{\theta}_{\text{old}}$), find $\boldsymbol{\theta}_{\text{new}}$ by maximizing L.

Policy-Based Reinforcement Learning

Policy-Based Reinforcement Learning

- Use policy network, $\pi(a|s;\boldsymbol{\theta})$, for controlling the agent.

Policy-Based Reinforcement Learning

- Use policy network, $\pi(a|s;\boldsymbol{\theta})$, for controlling the agent.
- State-value function:

$$V_\pi(s) = \mathbb{E}_{A\sim\pi}[Q_\pi(s,A)]$$

$$= \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a).$$

Policy-Based Reinforcement Learning

- Use policy network, $\pi(a|s;\boldsymbol{\theta})$, for controlling the agent.
- State-value function:

$$V_\pi(s) = \mathbb{E}_{A\sim\pi}[Q_\pi(s,A)]$$

$$= \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a).$$

- Objective function:

$$J(\boldsymbol{\theta}) = \mathbb{E}_S[V_\pi(S)].$$

Deriving Objective Function

- State-value function:

$$V_\pi(s) = \sum_a \pi(a \mid s; \boldsymbol{\theta}) \cdot Q_\pi(s, a)$$

Deriving Objective Function

- State-value function:

$$V_\pi(s) = \sum_a \pi(a \mid s; \boldsymbol{\theta}) \cdot Q_\pi(s, a)$$

$$= \sum_a \pi(a \mid s; \boldsymbol{\theta}_{\text{old}}) \cdot \frac{\pi(a \mid s; \boldsymbol{\theta})}{\pi(a \mid s; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(s, a)$$

Deriving Objective Function

- State-value function:

$$
\begin{aligned}
V_\pi(s) &= \textstyle\sum_a \pi(a \mid s; \boldsymbol{\theta}) \cdot Q_\pi(s, a) \\
&= \textstyle\sum_a \pi(a \mid s; \boldsymbol{\theta}_{\text{old}}) \cdot \frac{\pi(a \mid s; \boldsymbol{\theta})}{\pi(a \mid s; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(s, a) \\
&= \mathbb{E}_{A \sim \pi(\cdot \mid s; \boldsymbol{\theta}_{\text{old}})} \left[\frac{\pi(A \mid s; \boldsymbol{\theta})}{\pi(A \mid s; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(s, A) \right].
\end{aligned}
$$

Deriving Objective Function

- Objective function:

$$J(\boldsymbol{\theta}) = \mathbb{E}_S[V_\pi(S)]$$

Deriving Objective Function

- Objective function:

$$
\begin{aligned}
J(\boldsymbol{\theta}) &= \mathbb{E}_S\left[V_\pi(S)\right] \\
&= \mathbb{E}_S\left[\mathbb{E}_A\left[\frac{\pi(A \mid S; \boldsymbol{\theta})}{\pi(A \mid S; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(S, A)\right]\right].
\end{aligned}
$$

Trust Region Policy Optimization (TRPO)

Reference:

- Schulman, Levine, Abbeel, Jordan, & Moritz. Trust region policy optimization. In *ICML*, 2015.

Why TRPO?

- More robust than policy gradient algorithms.
- More sample efficient than policy gradient algorithms.

Overview of TRPO

Repeat the two steps:

1. **Approximation:** Given $\boldsymbol{\theta}_{\text{old}}$, we construct $L(\boldsymbol{\theta}|\boldsymbol{\theta}_{\text{old}})$ that approximates $J(\boldsymbol{\theta})$in the neighborhood of $\boldsymbol{\theta}_{\text{old}}$.

2. **Maximization:** In the trust region, $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$, find $\boldsymbol{\theta}_{\text{new}}$ by:

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \underset{\boldsymbol{\theta} \in \mathcal{N}(\boldsymbol{\theta}_{\text{old}})}{\operatorname{argmax}} \; L(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}).$$

Step 1: Approximation

Approximate $J(\boldsymbol{\theta})$ in the neighborhood of $\boldsymbol{\theta}_{\text{old}}$.

Step 1: Approximation

Objective function: $J(\boldsymbol{\theta}) = \mathbb{E}_{S,\,A}\left[\frac{\pi(A \mid S;\, \boldsymbol{\theta})}{\pi(A \mid S;\, \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(S, A)\right].$

Step 1: Approximation

Objective function: $J(\boldsymbol{\theta}) = \mathbb{E}_{S,A}\left[\frac{\pi(A \mid S; \boldsymbol{\theta})}{\pi(A \mid S; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(S, A)\right].$

- S is sampled from the **state transition** of the environment.
- A is sampled from the **policy** $\pi(A \mid s; \boldsymbol{\theta}_{\text{old}})$.

Step 1: Approximation

Objective function: $J(\boldsymbol{\theta}) = \mathbb{E}_{S,A}\left[\frac{\pi(A \mid S; \boldsymbol{\theta})}{\pi(A \mid S; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(S, A)\right].$

- Controlled by the policy, $\pi(A \mid s; \boldsymbol{\theta}_{\text{old}})$, the agent collects a trajectory:

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n.$$

Step 1: Approximation

Objective function: $J(\boldsymbol{\theta}) = \mathbb{E}_{S,A}\left[\frac{\pi(A \mid S; \boldsymbol{\theta})}{\pi(A \mid S; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(S, A)\right].$

- Controlled by the policy, $\pi(A \mid s; \boldsymbol{\theta}_{\text{old}})$, the agent collects a trajectory:

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n.$$

- Monte Carlo approximation to the expectation:

$$L(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(s_i, a_i).$$

Step 1: Approximation

Approximate $J(\boldsymbol{\theta})$ by $L(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(s_i, a_i)$.

Step 1: Approximation

Approximate $J(\boldsymbol{\theta})$ by $L(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(s_i, a_i)$.

- Observed rewards in an episode: $r_1, r_2, r_3, \cdots, r_n$.
- Discounted return:

$$u_i = r_i + \gamma \cdot r_{i+1} + \gamma^2 \cdot r_{i+2} + \cdots + \gamma^{n-i} \cdot r_n.$$

Step 1: Approximation

Approximate $J(\boldsymbol{\theta})$ by $L(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(s_i, a_i)$.

- Observed rewards in an episode: $r_1, r_2, r_3, \cdots, r_n$.
- Discounted return:

$$u_i = r_i + \gamma \cdot r_{i+1} + \gamma^2 \cdot r_{i+2} + \cdots + \gamma^{n-i} \cdot r_n.$$

- Monte Carlo approximation: $Q_\pi(s_i, a_i) \approx u_i$.

Step 1: Approximation

Approximate $J(\boldsymbol{\theta})$ by $L(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot Q_\pi(s_i, a_i)$.

$\approx u_i$

Approximate $J(\boldsymbol{\theta})$ by $\tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot u_i$.

Step 1: Approximation

Approximate $J(\boldsymbol{\theta})$ by $\tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot u_i$.

Step 2: Maximization

Approximate $J(\boldsymbol{\theta})$ by $\tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot u_i .$

- In the trust region, $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$, find $\boldsymbol{\theta}_{\text{new}}$ by:

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}); \quad \text{s.t.} \;\; \boldsymbol{\theta} \in \mathcal{N}(\boldsymbol{\theta}_{\text{old}}).$$

Step 2: Maximization

Approximate $J(\boldsymbol{\theta})$ by $\tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot u_i$.

- In the trust region, $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$, find $\boldsymbol{\theta}_{\text{new}}$ by:

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}); \quad \text{s.t.}\ \boldsymbol{\theta} \in \mathcal{N}(\boldsymbol{\theta}_{\text{old}}).$$

- **Option 1:** $\|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{old}}\| < \Delta.$

Step 2: Maximization

Approximate $J(\boldsymbol{\theta})$ by $\tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot u_i$.

- In the trust region, $\mathcal{N}(\boldsymbol{\theta}_{\text{old}})$, find $\boldsymbol{\theta}_{\text{new}}$ by:

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}})\,; \quad \text{s.t.}\ \boldsymbol{\theta} \in \mathcal{N}(\boldsymbol{\theta}_{\text{old}}).$$

- **Option 1:** $\left|\left|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{old}}\right|\right| < \Delta.$
- **Option 2:** $\frac{1}{n}\sum_{i=1}^{n} \mathrm{KL}\left[\pi(\cdot \mid s_i; \boldsymbol{\theta}_{\text{old}}) \,||\, \pi(\cdot \mid s_i; \boldsymbol{\theta})\right] < \Delta.$

TRPO: Summary

1. Controlled by the policy, $\pi(\cdot \mid s; \boldsymbol{\theta}_{\text{old}})$, the agent plays a game to the end and observes a trajectory:

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n.$$

2. For $i = 1, \cdots, n$, compute discounted returns: $u_i = \sum_{k=i}^{n} \gamma^{k-i} \cdot r_k$.

TRPO: Summary

1. Controlled by the policy, $\pi(\cdot \mid s; \boldsymbol{\theta}_{\text{old}})$, the agent plays a game to the end and observes a trajectory:

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n.$$

2. For $i = 1, \cdots, n$, compute discounted returns: $u_i = \sum_{k=i}^{n} \gamma^{k-i} \cdot r_k$.

3. **Approximation:** $\tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot u_i$.

TRPO: Summary

1. Controlled by the policy, $\pi(\cdot \mid s; \boldsymbol{\theta}_{\text{old}})$, the agent plays a game to the end and observes a trajectory:

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n.$$

2. For $i = 1, \cdots, n$, compute discounted returns: $u_i = \sum_{k=i}^{n} \gamma^{k-i} \cdot r_k$.

3. **Approximation:** $\tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(a_i \mid s_i; \boldsymbol{\theta})}{\pi(a_i \mid s_i; \boldsymbol{\theta}_{\text{old}})} \cdot u_i$.

4. **Maximization:**

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ \tilde{L}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{old}}); \quad \text{s.t.}\ \left\|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{old}}\right\| < \Delta.$$

Policy Gradient versus TRPO

- They are both policy-based reinforcement learning; they have the same objective function:

$$J(\boldsymbol{\theta}) = \mathbb{E}_S[V_\pi(S)].$$

- Policy gradient algorithms maximize $J(\boldsymbol{\theta})$ by stochastic gradient ascent.

- TRPO maximizes $J(\boldsymbol{\theta})$ by trust-region algorithm.

Thank you!