We present a method for **temporally-extended planning** over high-dimensional state spaces by learning a state representation amenable to optimization and a goal-conditioned policy to abstract time.

**Introduction**
- Planning can solve temporally extended tasks
- Goals provide action and temporal abstraction [2]
- Generally, states may live in unknown lower-dimensional manifold, making planning challenging

**Idea:** Learn dense state abstractions to make optimization feasible.

**Temporal difference models** (TDMs) are value functions that satisfy

\[ V(s_t, g, t) = \mathbb{E}[r(s_t, a_t, s_{t+1}, s_g)1\{t = 0\} + \max Q(s_{t+1}, a, s_g, t - 1)1\{t \neq 0\}] \]

\[ r(s_t, a_t, s_{t+1}, g) = -\text{Distance}(s_{t+1}, g) \]

where \( t \) = planning horizon and \( s_g \) = goal state.

[Image 0x64 to 84x173]

**Latent Embedding for Abstracted Planning (LEAP)**

Given the current state \( s \) and goal \( g \)

choose realistic latent vectors \( z_1, \ldots, z_k \)

that minimize the norm of the feasibility vector

\[ \overline{V}(s, g; t_{1:k}, g) = \begin{bmatrix} V(s, g_1, t_1) \\ V(g_1, g_2, t_1) \\ \vdots \\ V(g_{t-1}, g_t) \end{bmatrix} \]

where \( g_k = \psi(z_k) \). Formally, minimize

\[ z_{1:k}^* = \arg \min_{z} \sum_{k=1}^{K} \| \overline{V}(s, z_{1:k}, t_{1:k}, g) \|_p - \lambda \sum_{k=1}^{K} \log p(z_k) \]

and go towards first goal \( g_1 = \psi(z_1^*) \).

**Implementation details**
- Use cross-entropy method for optimization.
- Reuse encoder \( \psi \) for RL networks.
- Use \( \ell_{\infty} \) norm.
- Uniformly space \( t_1, t_2, \ldots, t_K = \lfloor T_{\text{max}} / K \rfloor \)

**References**
