Objective Geometry: Modeling Goals as Landscapes in High-Dimensional Policy Space

Abstract

We formalize objectives as continuous functions $J : \Theta \to \mathbb{R}$ over policy space $\Theta \subseteq \mathbb{R}^n$. The gradient field $\nabla_\theta J$ defines optimization dynamics. Safety regions $S \subset \Theta$ are subsets with invariant properties under gradient flow. Landscape shaping operators $T : (\Theta \to \mathbb{R}) \to (\Theta \to \mathbb{R})$ modify objective structure to preserve safety invariants. We characterize conditions for safety region invariance (Proposition 1), phase transitions in landscape topology (Proposition 2), and gradient aliasing (Proposition 3). Failure modes include geometric collapse (proxy objectives with different landscape structure), boundary tunneling (optimization escaping S), and gradient aliasing (distinct objectives producing identical gradients). Control is achieved through landscape modification operators that preserve desired invariants. Computational complexity scales with $\dim(\Theta)$ and landscape exploration requirements.

1 Formal Definitions

Definition 1 (Policy Space). *Let $\Theta \subseteq \mathbb{R}^n$ denote the space of all possible policies, where $n \in \mathbb{N}$ is the dimensionality. Each $\theta \in \Theta$ is a policy parameterization.*

Definition 2 (Objective Function). *An objective function is a mapping $J : \Theta \to \mathbb{R}$. The pair (Θ, J) defines an objective landscape.*

Definition 3 (Gradient Field). *The gradient field of J is $\nabla_\theta J : \Theta \to \mathbb{R}^n$ where $(\nabla_\theta J(\theta))_i = \frac{\partial J}{\partial \theta_i}(\theta)$.*

Definition 4 (Gradient Flow). *A gradient flow is a trajectory $\theta : [0, T] \to \Theta$ satisfying:*

$$\frac{d\theta}{dt} = -\alpha \nabla_\theta J(\theta(t))$$

where $\alpha > 0$ is the learning rate parameter.

Definition 5 (Basin). *A basin $B \subseteq \Theta$ is a connected component of the sublevel set $\{\theta \in \Theta : J(\theta) \leq c\}$ for some $c \in \mathbb{R}$, such that B contains a local minimum θ^* where $\nabla_\theta J(\theta^*) = 0$ and $\nabla^2_\theta J(\theta^*)$ is positive definite.*

Definition 6 (Optimization Cliff). *An optimization cliff is a region $C \subseteq \Theta$ where $\|\nabla_\theta J(\theta)\| > \tau$ for all $\theta \in C$, where $\tau > 0$ is a threshold parameter.*

Definition 7 (Barrier). *A barrier $B \subseteq \Theta$ is a region such that for all $\theta \in B$, there exists $\epsilon > 0$ such that for all $\theta' \in B_\epsilon(\theta) \setminus B$, we have $J(\theta') > J(\theta)$, where $B_\epsilon(\theta)$ is the ϵ-ball around θ.*

Definition 8 (Safety Region). *A safety region $S \subseteq \Theta$ is a closed set with the property that if $\theta(0) \in S$ and θ is a gradient flow, then $\theta(t) \in S$ for all $t \geq 0$ (forward invariance).*

Definition 9 (Landscape Shaping Operator)**.** *A landscape shaping operator is a mapping $T : (\Theta \to \mathbb{R}) \to (\Theta \to \mathbb{R})$ that transforms objective functions. The modified objective is $J' = T(J)$.*

Definition 10 (Invariant Preservation)**.** *A landscape shaping operator T preserves invariant $P : (\Theta \to \mathbb{R}) \to \{true, false\}$ if for all $J : \Theta \to \mathbb{R}$, $P(J) \Rightarrow P(T(J))$.*

Definition 11 (Geometric Collapse)**.** *Geometric collapse occurs when objectives $J_1, J_2 : \Theta \to \mathbb{R}$ satisfy $J_1(\theta^*) = J_2(\theta^*)$ for some $\theta^* \in \Theta$, but the basins B_1, B_2 containing θ^* under J_1, J_2 respectively have $B_1 \neq B_2$.*

Definition 12 (Boundary Tunneling)**.** *Boundary tunneling occurs when a safety region S and gradient flow θ satisfy $\theta(0) \in S$ but $\theta(t) \notin S$ for some $t > 0$, violating forward invariance.*

Definition 13 (Gradient Aliasing)**.** *Gradient aliasing occurs when distinct objectives $J_1, J_2 : \Theta \to \mathbb{R}$ satisfy $\nabla_\theta J_1(\theta) = \nabla_\theta J_2(\theta)$ for all θ in some region $R \subseteq \Theta$, but $J_1 \neq J_2$ on R.*

2 Propositions

Proposition 1 (Safety Region Invariance)**.** *Let $S \subseteq \Theta$ be a closed set and $J : \Theta \to \mathbb{R}$ be continuously differentiable. Then S is a safety region (forward invariant under gradient flow) if and only if for all $\theta \in \partial S$ (the boundary of S), the gradient $\nabla_\theta J(\theta)$ points into S or is tangent to ∂S.*

Formally: S is forward invariant if and only if $\forall \theta \in \partial S, \forall \mathbf{n} \in N_S(\theta)$ (outward normal vectors), we have $\mathbf{n} \cdot \nabla_\theta J(\theta) \geq 0$.

Proof. Forward invariance requires that gradient flow cannot cross ∂S from inside to outside. At boundary points, the gradient component normal to the boundary must be non-positive (pointing inward or tangent). This is equivalent to the condition stated. □

Proposition 2 (Phase Transition Conditions)**.** *Let $J_\lambda : \Theta \to \mathbb{R}$ be a family of objectives parameterized by $\lambda \in \mathbb{R}$. A phase transition occurs at λ_0 if the number of basins changes discontinuously: $\lim_{\lambda \to \lambda_0^-} |\mathcal{B}_\lambda| \neq \lim_{\lambda \to \lambda_0^+} |\mathcal{B}_\lambda|$, where $\mathcal{B}_\lambda$ is the set of basins under J_λ.*

Necessary condition: There exists $\theta^* \in \Theta$ such that $\nabla_\theta J_{\lambda_0}(\theta^*) = 0$ and $\det(\nabla^2_\theta J_{\lambda_0}(\theta^*)) = 0$ (a degenerate critical point).

Proof. Basins are defined by local minima. A change in the number of basins requires a change in the number of local minima. This occurs when a critical point becomes degenerate (Hessian loses full rank), allowing minima to merge, split, or disappear. □

Proposition 3 (Gradient Aliasing Conditions)**.** *Let $J_1, J_2 : \Theta \to \mathbb{R}$ be continuously differentiable objectives. If $\nabla_\theta J_1(\theta) = \nabla_\theta J_2(\theta)$ for all $\theta \in R \subseteq \Theta$ where R is connected, then $J_1(\theta) = J_2(\theta) + c$ for some constant $c \in \mathbb{R}$ and all $\theta \in R$.*

Corollary: True gradient aliasing (identical gradients but non-constant difference) can only occur if R is not connected or if the objectives are not continuously differentiable.

Proof. If gradients are identical on a connected region, then $J_1 - J_2$ has zero gradient on that region. By the mean value theorem, $J_1 - J_2$ is constant on connected components. If R is connected, the difference is constant. □

Proposition 4 (Barrier Effectiveness)**.** *Let $B \subseteq \Theta$ be a barrier under objective J, and let θ be a gradient flow with $\theta(0) \notin B$. If $\|\nabla_\theta J(\theta)\| > 0$ for all $\theta \in \partial B$ and the barrier is sufficiently "high" (there exists $\delta > 0$ such that $J(\theta') - J(\theta) > \delta$ for all $\theta \in B, \theta' \in B_\epsilon(\theta) \setminus B$), then $\theta(t) \notin B$ for all $t \geq 0$ provided α (learning rate) is sufficiently small.*

Proof. The gradient flow cannot cross the barrier if the objective increase required to cross exceeds what can be achieved in a single step. For small α, the step size $|\alpha \nabla_\theta J(\theta)|$ is bounded, so if the barrier height exceeds this bound, crossing is impossible. □

3 Failure Modes

3.1 Goodhart via Geometric Collapse

Formalization: Let $J_{\text{proxy}} : \Theta \to \mathbb{R}$ be a proxy objective and $J_{\text{true}} : \Theta \to \mathbb{R}$ be the true objective. Geometric collapse occurs when:

$$\exists \theta^* \in \Theta : J_{\text{proxy}}(\theta^*) = \max_{\theta \in \Theta} J_{\text{proxy}}(\theta) \wedge J_{\text{true}}(\theta^*) \neq \max_{\theta \in \Theta} J_{\text{true}}(\theta)$$

Moreover, the basins $B_{\text{proxy}}, B_{\text{true}}$ containing the respective optima satisfy $B_{\text{proxy}} \cap B_{\text{true}} = \emptyset$ or have significantly different structure.

Mechanism: Optimization under J_{proxy} follows gradient flow into B_{proxy}, but this basin does not correspond to high values under J_{true}. The landscape structure differs between proxy and true objectives.

Conditions: Geometric collapse is more likely when:

- $\|\nabla_\theta J_{\text{proxy}} - \nabla_\theta J_{\text{true}}\|$ is large in regions of interest
- The number of basins differs: $|\mathcal{B}_{\text{proxy}}| \neq |\mathcal{B}_{\text{true}}|$
- Basin boundaries do not align: $\partial B_{\text{proxy}} \not\approx \partial B_{\text{true}}$

3.2 Boundary Tunneling

Formalization: Let $S \subseteq \Theta$ be a safety region and θ be a gradient flow. Boundary tunneling occurs when:

$$\theta(0) \in S \wedge \exists t > 0 : \theta(t) \notin S$$

Mechanism: The gradient field $\nabla_\theta J$ points outward from S at boundary points, or the gradient magnitude is large enough that discrete optimization steps (with finite learning rate α) overshoot the boundary.

Conditions: Boundary tunneling occurs when Proposition 1 (safety region invariance) fails:

- $\exists \theta \in \partial S, \exists \mathbf{n} \in N_S(\theta) : \mathbf{n} \cdot \nabla_\theta J(\theta) < 0$
- The learning rate α is large enough that $|\alpha \nabla_\theta J(\theta)|$ exceeds the distance to ∂S
- The boundary ∂S has high curvature, creating "tunnels" where gradient flow can escape

3.3 Gradient Aliasing

Formalization: Gradient aliasing occurs when:

$$\exists J_1, J_2 : \Theta \to \mathbb{R}, \exists R \subseteq \Theta : J_1 \neq J_2 \text{ on } R \wedge \nabla_\theta J_1(\theta) = \nabla_\theta J_2(\theta) \forall \theta \in R$$

By Proposition 3, this requires either R is not connected or the objectives violate smoothness assumptions.

Mechanism: Multiple distinct objectives produce identical optimization dynamics in region R. Optimization cannot distinguish between J_1 and J_2 based on gradient information alone.

Implications:

- Optimization paths are identical under J_1 and J_2 in R
- Safety properties that depend on objective values (not just gradients) may differ
- Landscape shaping that modifies J_1 may not affect optimization if aliasing occurs

Special case: When $J_1(\theta) = J_2(\theta) + c$ on connected R, the constant offset c does not affect gradients but may affect safety region definitions if safety depends on absolute objective values.

4 Control Implications

4.1 Landscape Modification Operators

Landscape shaping is implemented through operators $T : (\Theta \to \mathbb{R}) \to (\Theta \to \mathbb{R})$. Common operators:

Barrier operator: $T_{\text{barrier}}(J)(\theta) = \begin{cases} J(\theta) + \lambda \cdot \text{penalty}(\theta) & \text{if } \theta \in D \\ J(\theta) & \text{otherwise} \end{cases}$

where $D \subseteq \Theta$ is a dangerous region and $\lambda > 0$ controls barrier height.

Smoothing operator: $T_{\text{smooth}}(J)(\theta) = \int_\Theta J(\theta')K(\theta, \theta')d\theta'$ where K is a smoothing kernel.

Basin widening operator: $T_{\text{widen}}(J)(\theta) = \min_{\theta' \in B_\epsilon(\theta)} J(\theta')$ for θ in safe regions, effectively taking the lower envelope.

4.2 Invariant Preservation

Landscape shaping must preserve desired properties. Let $P : (\Theta \to \mathbb{R}) \to \{\text{true}, \text{false}\}$ be a property (e.g., "safety region S is forward invariant").

Requirement: $P(J) \Rightarrow P(T(J))$ for landscape shaping operator T.

Verification: For safety region invariance (Proposition 1), verify that boundary conditions are preserved:

$$\forall \theta \in \partial S : \mathbf{n} \cdot \nabla_\theta T(J)(\theta) \geq 0 \text{ for all outward normals } \mathbf{n}$$

Sufficient condition: If T is monotone ($J_1 \leq J_2 \Rightarrow T(J_1) \leq T(J_2)$) and preserves gradient directions at boundaries, then forward invariance is preserved.

4.3 Constraint-Based Control

Safety regions can be enforced through constraints. The constrained optimization problem:

$$\min_{\theta \in \Theta} J(\theta) \text{ subject to } \theta \in S$$

defines an effective objective $J_S(\theta) = \begin{cases} J(\theta) & \text{if } \theta \in S \\ +\infty & \text{otherwise} \end{cases}$

This creates an infinite barrier, ensuring $\theta(t) \in S$ for all t if $\theta(0) \in S$ and the constraint is properly handled (e.g., via projection or barrier methods).

4.4 Computational Complexity

Landscape shaping operations have complexity that depends on:

- *Dimensionality*: Operations scale with $\dim(\Theta) = n$. Gradient computation: $O(n)$, barrier evaluation: $O(n)$ per point.
- *Landscape exploration*: Identifying basins, boundaries, and dangerous regions requires sampling. Worst case: exponential in n.
- *Operator application*: Smoothing and other integral operators require integration over Θ, typically approximated via sampling: $O(m \cdot n)$ where m is sample count.
- *Verification*: Checking safety region invariance requires evaluating boundary conditions. If ∂S has complexity $O(k)$, verification is $O(k \cdot n)$.

Practical limitation: Exhaustive landscape exploration is infeasible for large n. Approximation methods (sampling, dimensionality reduction, learned models) are necessary but introduce uncertainty.

5 Limits

5.1 Non-Claims

This framework does not address:

- *Value specification*: Defining what constitutes "safety" or "desired behavior" is outside scope. The framework assumes safety regions S are given.
- *Human cognition*: The geometric model applies to gradient-based optimization. Whether human goal-directed behavior follows similar dynamics is not addressed.
- *Multi-agent systems*: Interactions between multiple optimizing agents are not modeled.
- *Non-stationary environments*: Objectives that change during optimization (e.g., due to environment feedback) require extensions.
- *Discrete action spaces*: The continuous optimization model may not apply directly to discrete decision problems, though similar geometric ideas might hold.

5.2 Computational Bounds

Landscape exploration: Identifying all basins in $\Theta \subseteq \mathbb{R}^n$ requires, in worst case, exponential samples in n. Approximation is necessary.

Gradient computation: Computing $\nabla_\theta J(\theta)$ requires $O(n)$ function evaluations (via finite differences) or automatic differentiation with similar cost.

Safety verification: Verifying forward invariance of safety region S requires checking boundary conditions. If ∂S has k faces/regions, verification is $O(k \cdot n)$ per check.

Landscape shaping: Applying operator T typically requires evaluating J at multiple points. If T requires m evaluations, cost is $O(m \cdot \mathrm{cost}(J))$.

5.3 Dimensionality Limitations

In high dimensions ($n \gg 1$):

- *Basin structure*: Basins may not be "basin-like" in any low-dimensional sense. Geometry may be counterintuitive.
- *Optimization paths*: Gradient flows may exhibit unexpected behaviors (slow convergence, chaotic dynamics, convergence to unexpected regions).
- *Sampling*: Uniform sampling of Θ becomes infeasible. Concentration of measure effects dominate.
- *Visualization*: Direct visualization is impossible. Low-dimensional projections may be misleading.

Implication: Geometric intuitions from low dimensions may not apply. Formal analysis and verification become more important than visualization.

5.4 Smoothness Assumptions

The framework assumes J is continuously differentiable. If J is non-smooth:

- Gradients may not exist everywhere
- Gradient flows may not be well-defined
- Safety region invariance conditions (Proposition 1) require modification
- Landscape shaping operators must handle discontinuities

Extension: Subgradient methods and non-smooth analysis can extend the framework, but propositions require modification.

6 Termination

The Objective Geometry framework provides a formal structure for reasoning about objectives as geometric entities in policy space. Safety is characterized through invariant properties of regions under gradient flow. Control is achieved via landscape modification operators that preserve desired invariants.

Failure modes (geometric collapse, boundary tunneling, gradient aliasing) are formally defined and their conditions characterized. Computational complexity scales with dimensionality and landscape exploration requirements.

Limitations include: value specification is outside scope, computational costs may be prohibitive in high dimensions, and smoothness assumptions may not hold in practice.

The framework does not solve alignment but provides a formal language for reasoning about objective structure and optimization dynamics.