Threshold Strategy for a Leaking Corner-Free Hamilton-Jacobi Reachability with Decomposed Computations

Chong He, Mugilan Mariappan, Keval Vora and Mo Chen

Chong He: chong_he@sfu.ca

2025/12/10

Contents

- What is the “leaking corner issue”?
- Why is solving this issue important?
- How to solve this issue?

Project webpage

Application Example

Self-contained Subsystem Decomposition

2D Single Integrator

2D System:

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = u = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

1D Sub-system1:

$$\dot{x}_1 = \dot{x} = w_1 = u_x$$

1D Sub-system2:

$$\dot{x}_2 = \dot{y} = w_2 = u_y$$

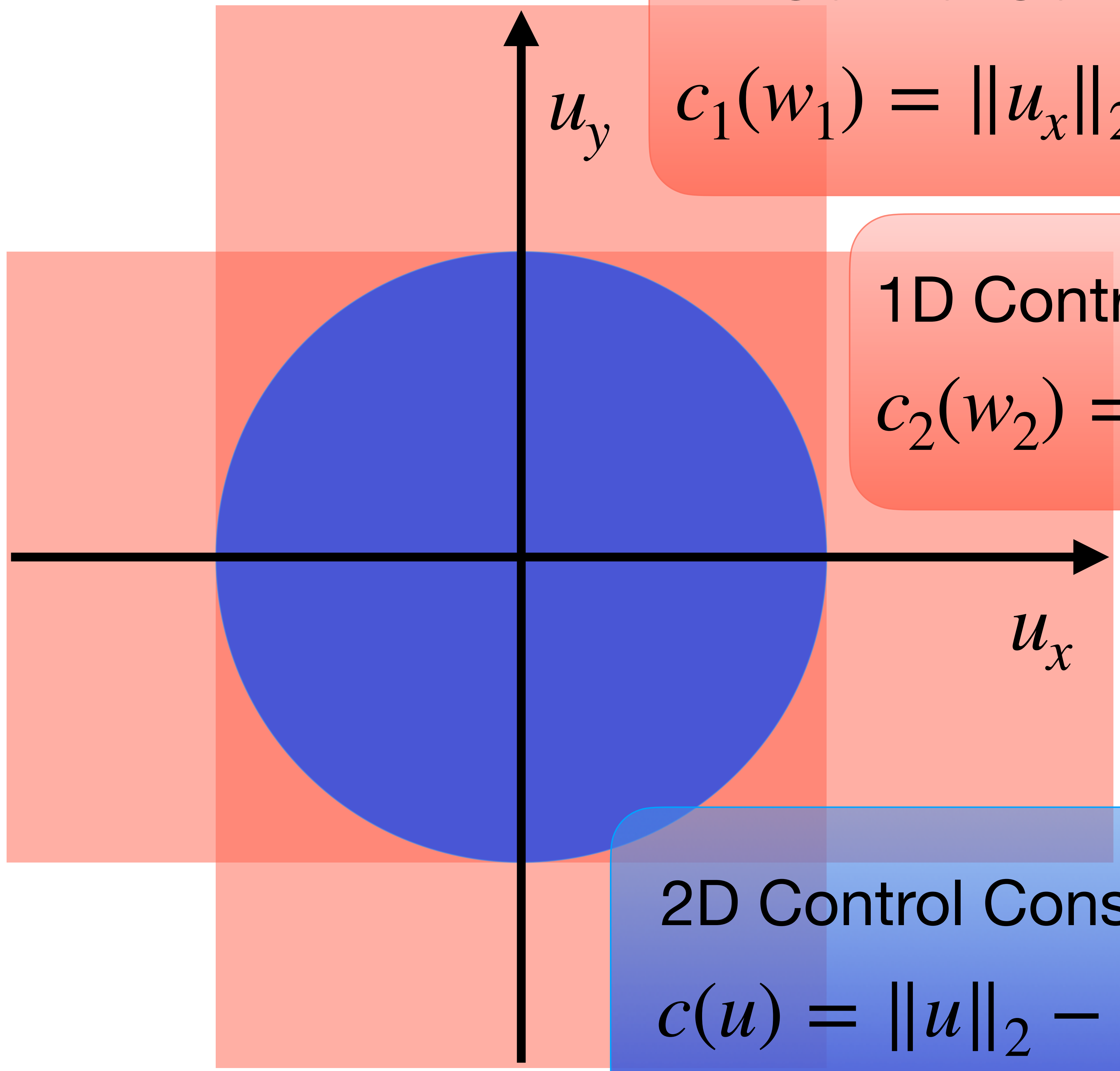
$$c_{\text{joint}}(w_1, w_2) = c(u) \leq 0$$

1D Control Constraint 1:

$$c_1(w_1) = \|u_x\|_2 - \bar{u} \leq 0$$

1D Control Constraint 2:

$$c_2(w_2) = \|u_y\|_2 - \bar{u} \leq 0$$



2D Control Constraint:

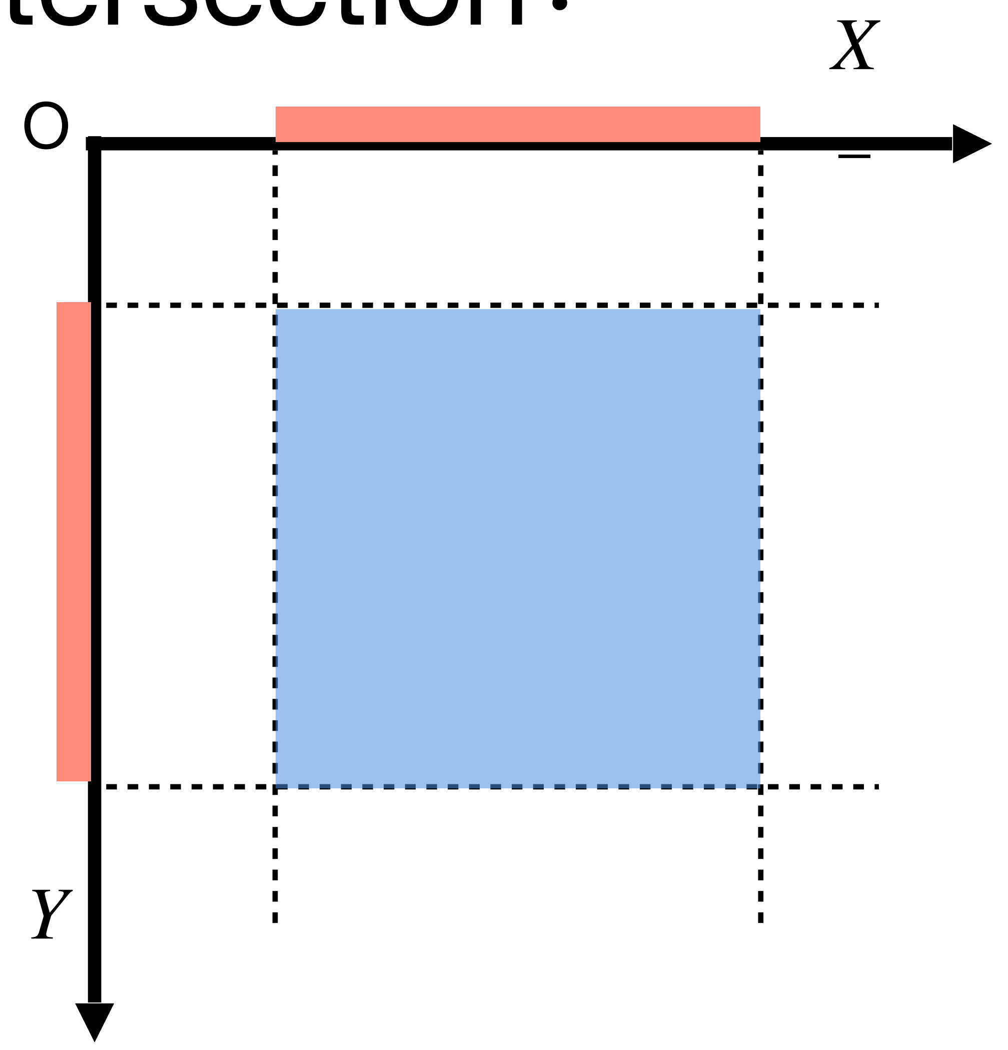
$$c(u) = \|u\|_2 - \bar{u} \leq 0$$

[1] Chen, M., Herbert, S. L., Vashishtha, M. S., Bansal, S., & Tomlin, C. J. (2018). Decomposition of reachable sets and tubes for a class of nonlinear systems. *IEEE Transactions on Automatic Control*, *63*(11), 3675-3688.

Application Example

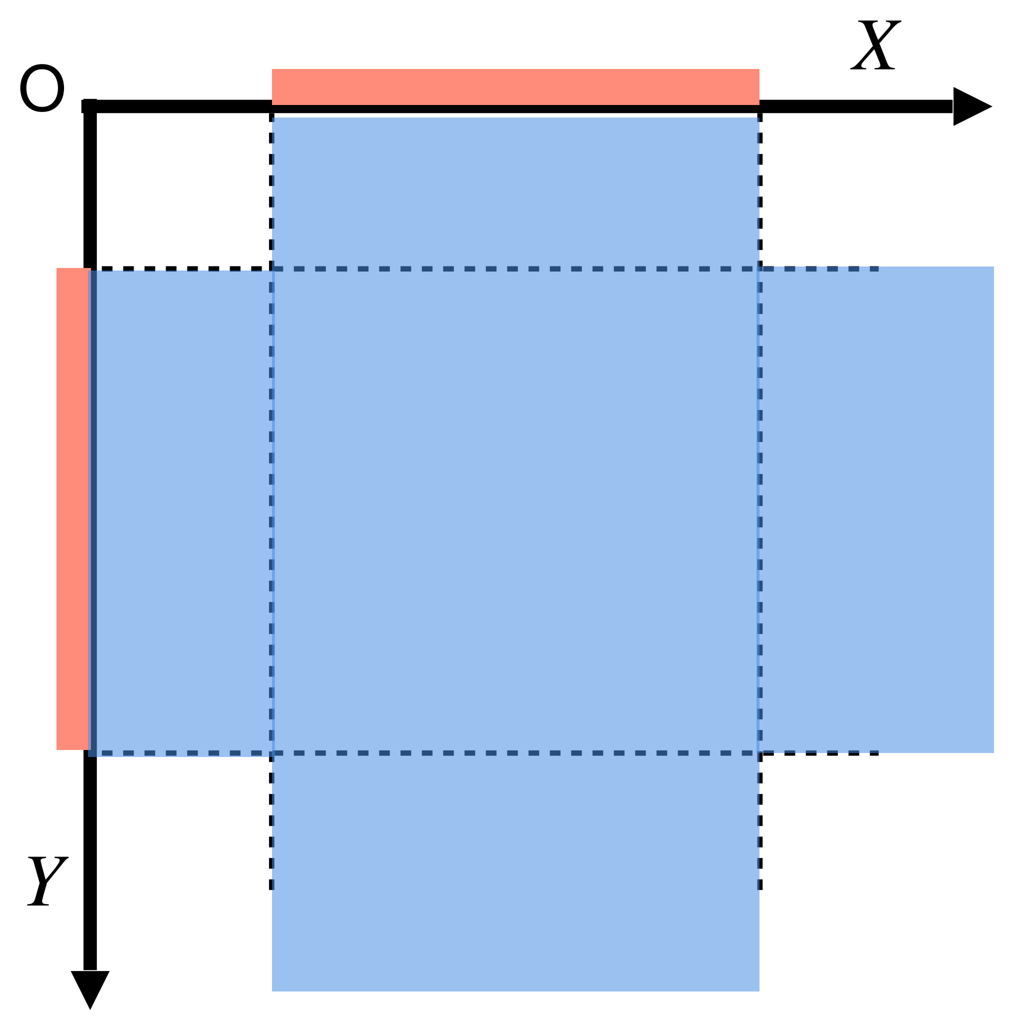
Value Function Decomposition

Intersection:



$$V(z, 0) = \max\{V_1(z, 0), V_2(z, 0)\}$$

Union:



$$V(z, 0) = \min\{V_1(z, 0), V_2(z, 0)\}$$

Full-dimensional sub-value function

Approximated Value Function:

Intersection: $\hat{V}(z, t) = \max\{V_1(z, t), V_2(z, t)\}$

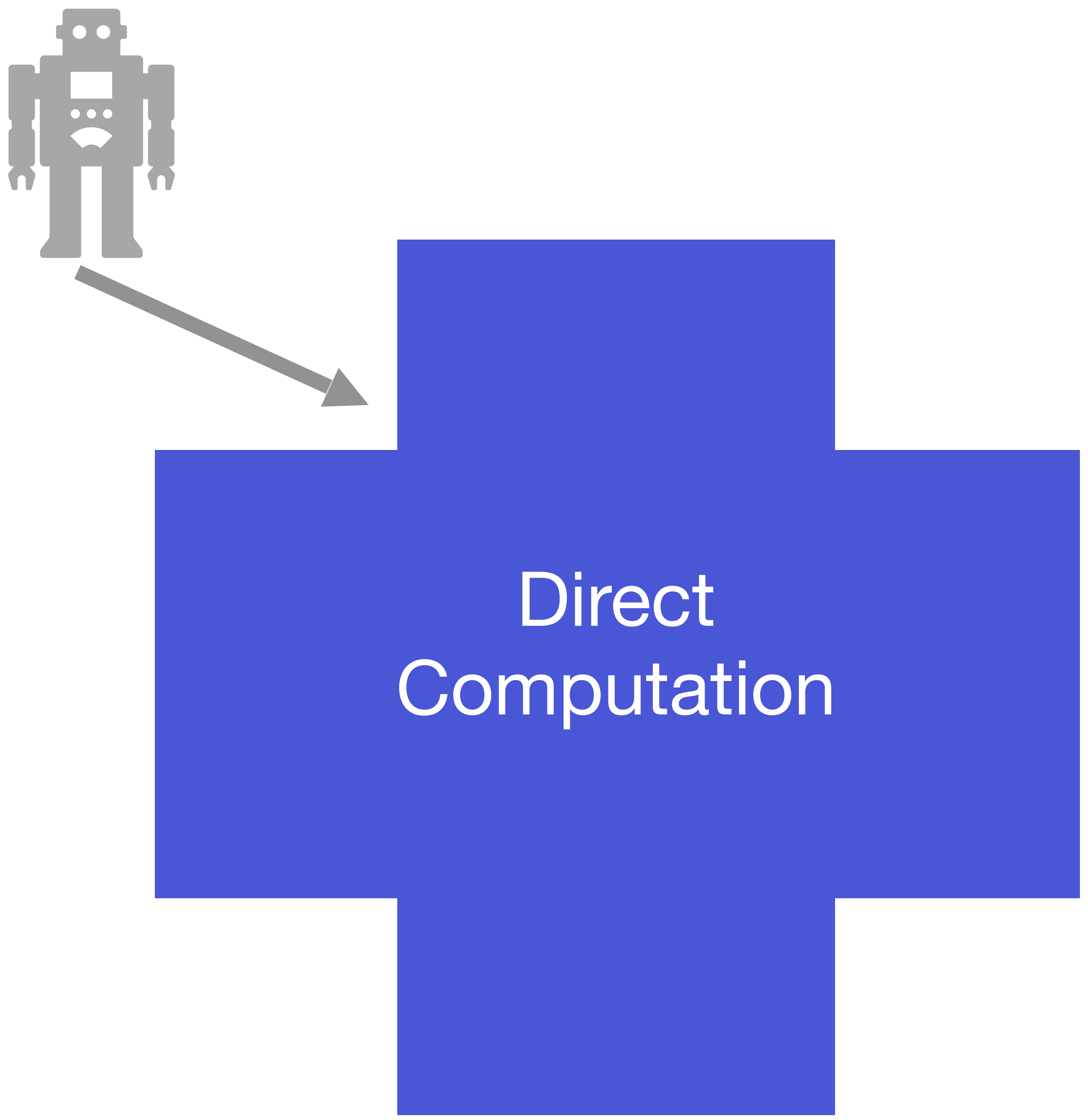
Union: $\hat{V}(z, t) = \min\{V_1(z, t), V_2(z, t)\}$

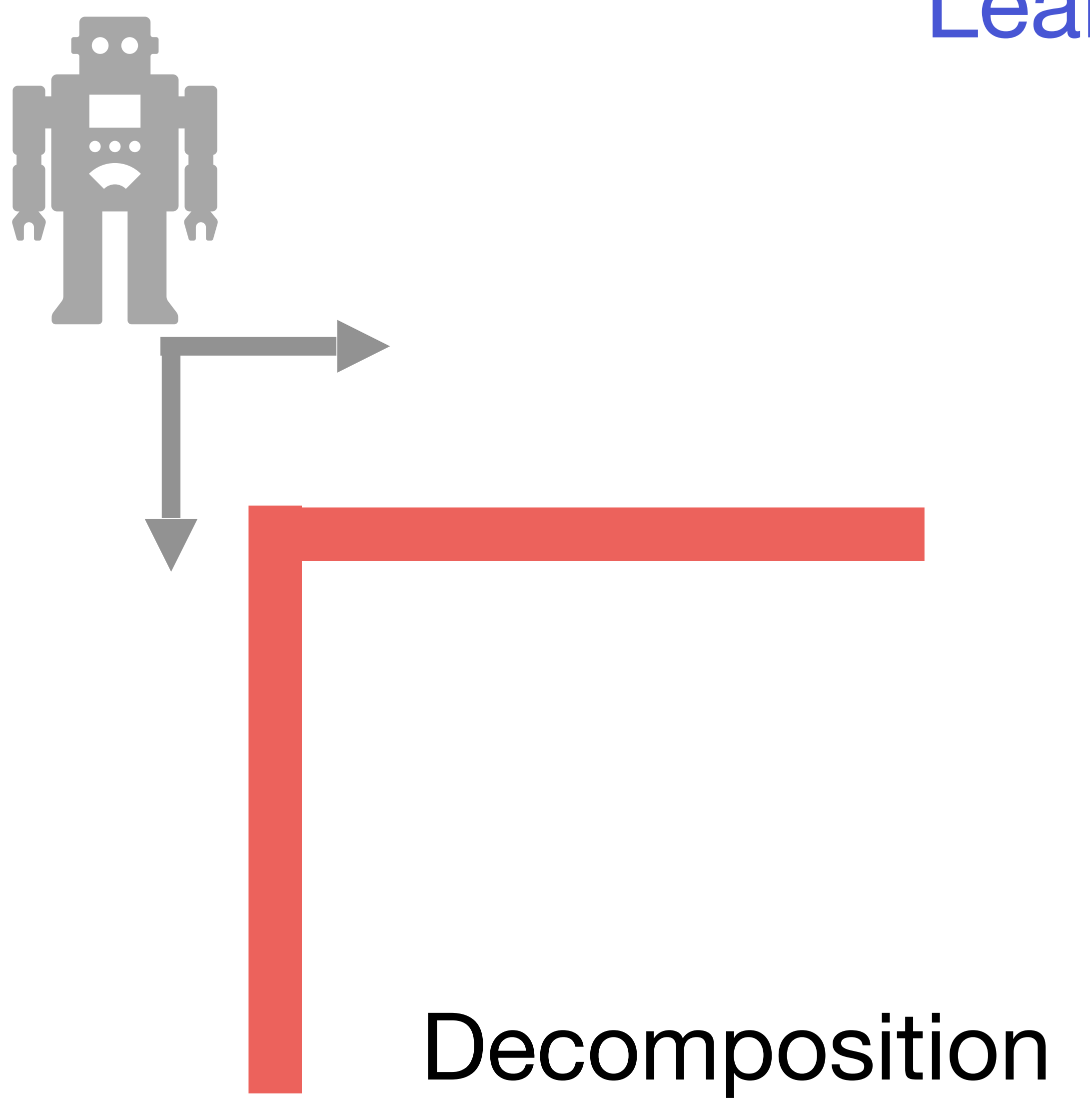
[1] Chen, M., Herbert, S. L., Vashishtha, M. S., Bansal, S., & Tomlin, C. J. (2018). Decomposition of reachable sets and tubes for a class of nonlinear systems. *IEEE Transactions on Automatic Control*, *63*(11), 3675-3688.

Leaking Corner Issue

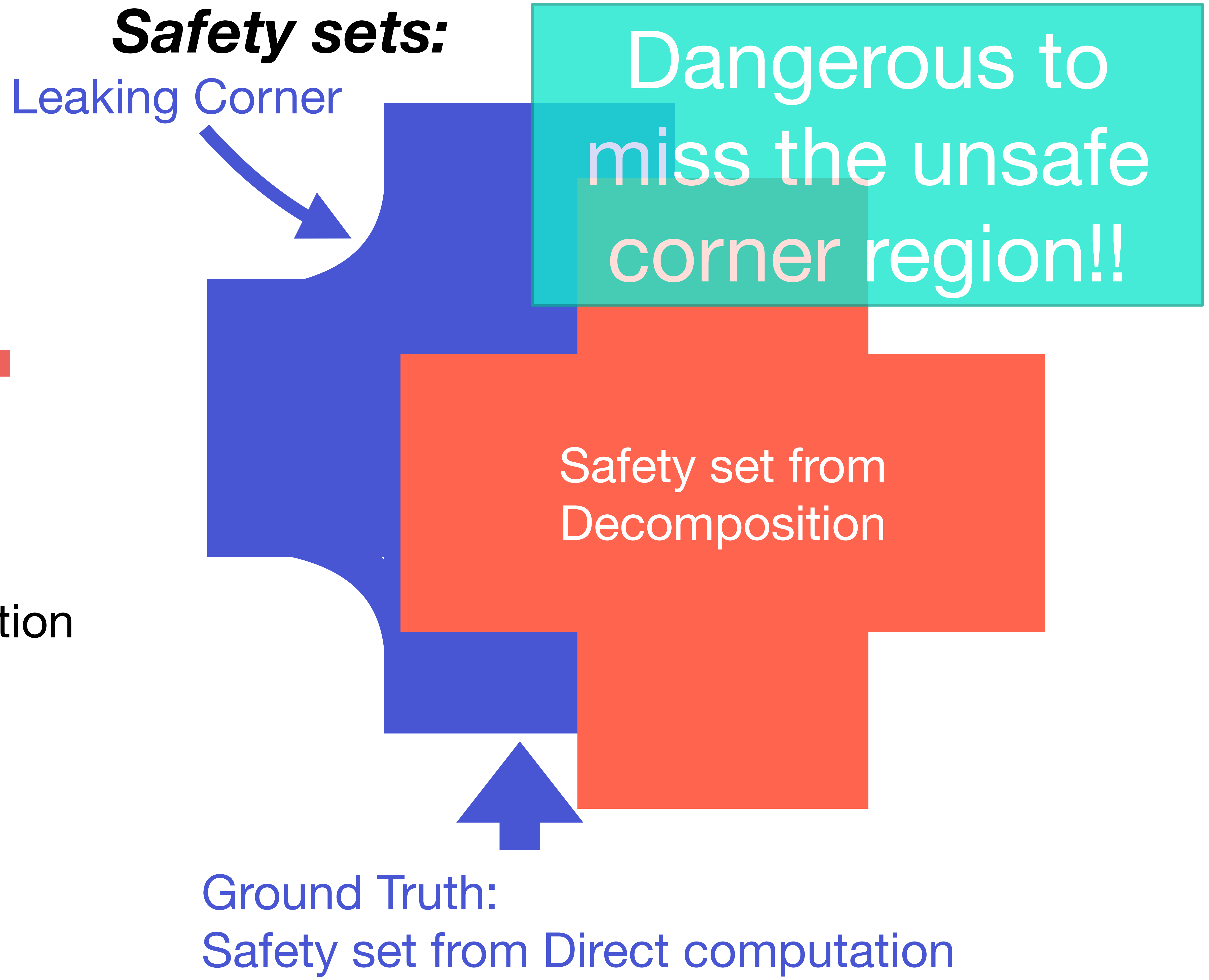
- When the low-dimensional control are constrained with each other

Regions to avoid:



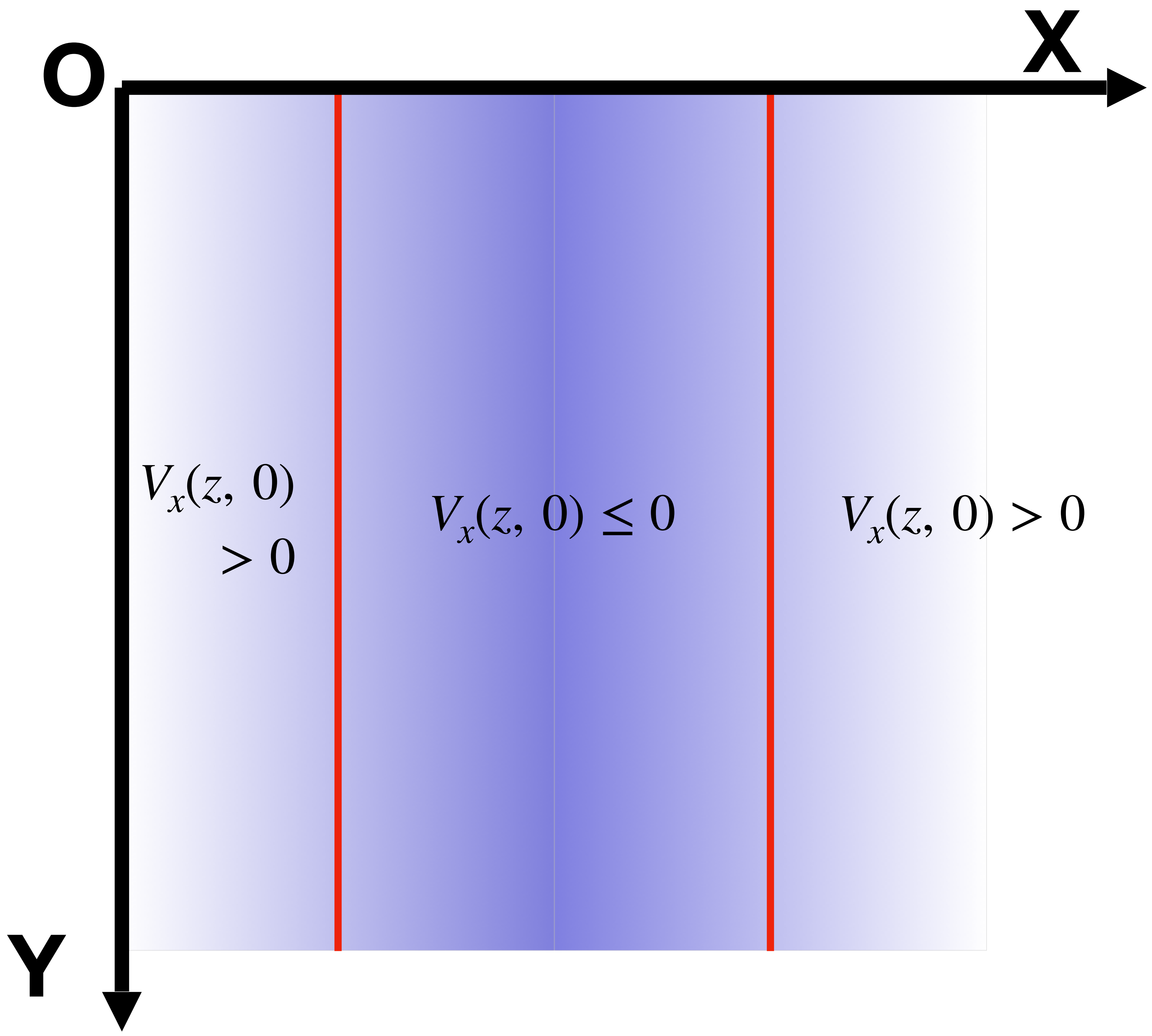


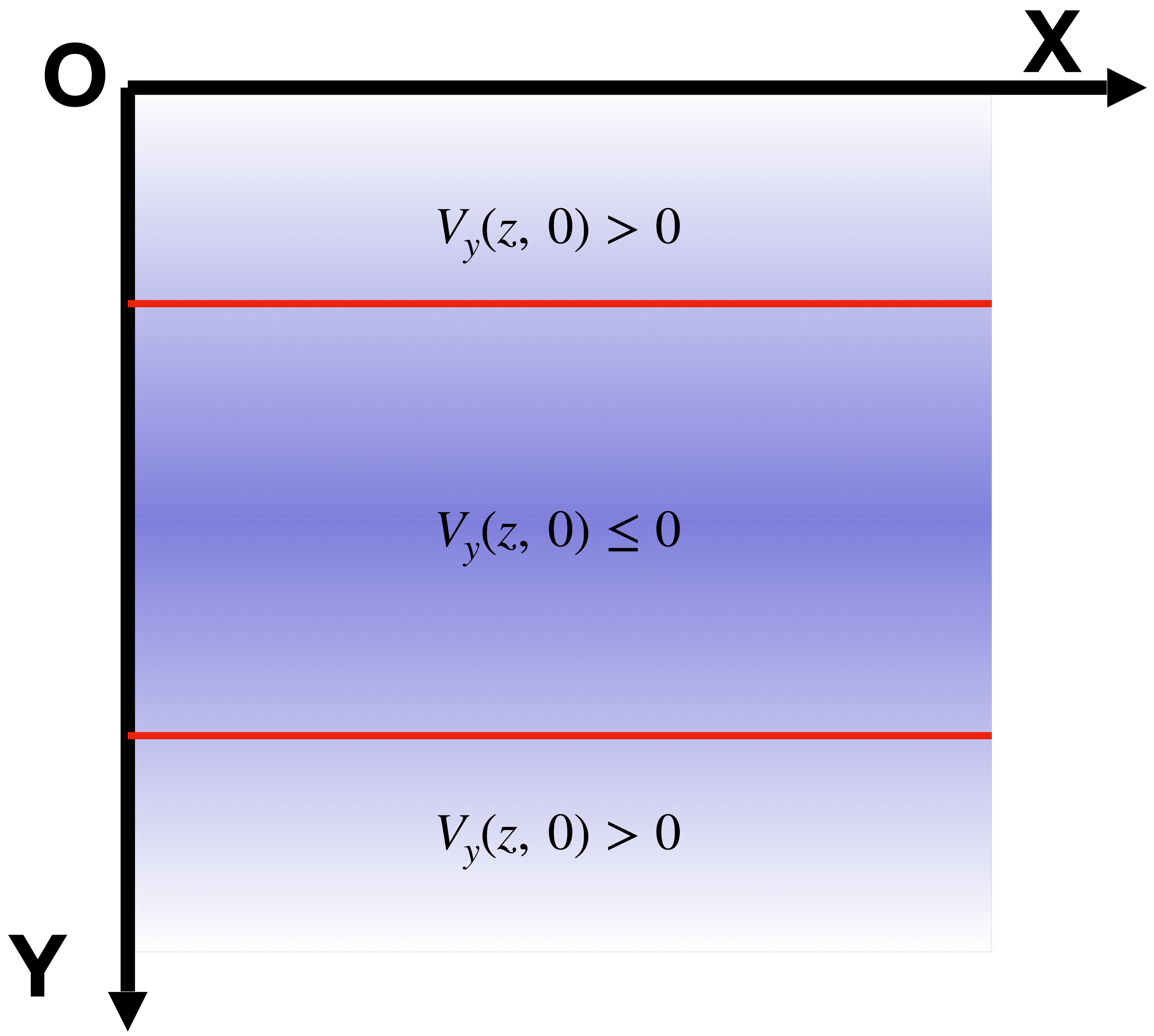
Safety sets:



Full-dimensional sub-value functions

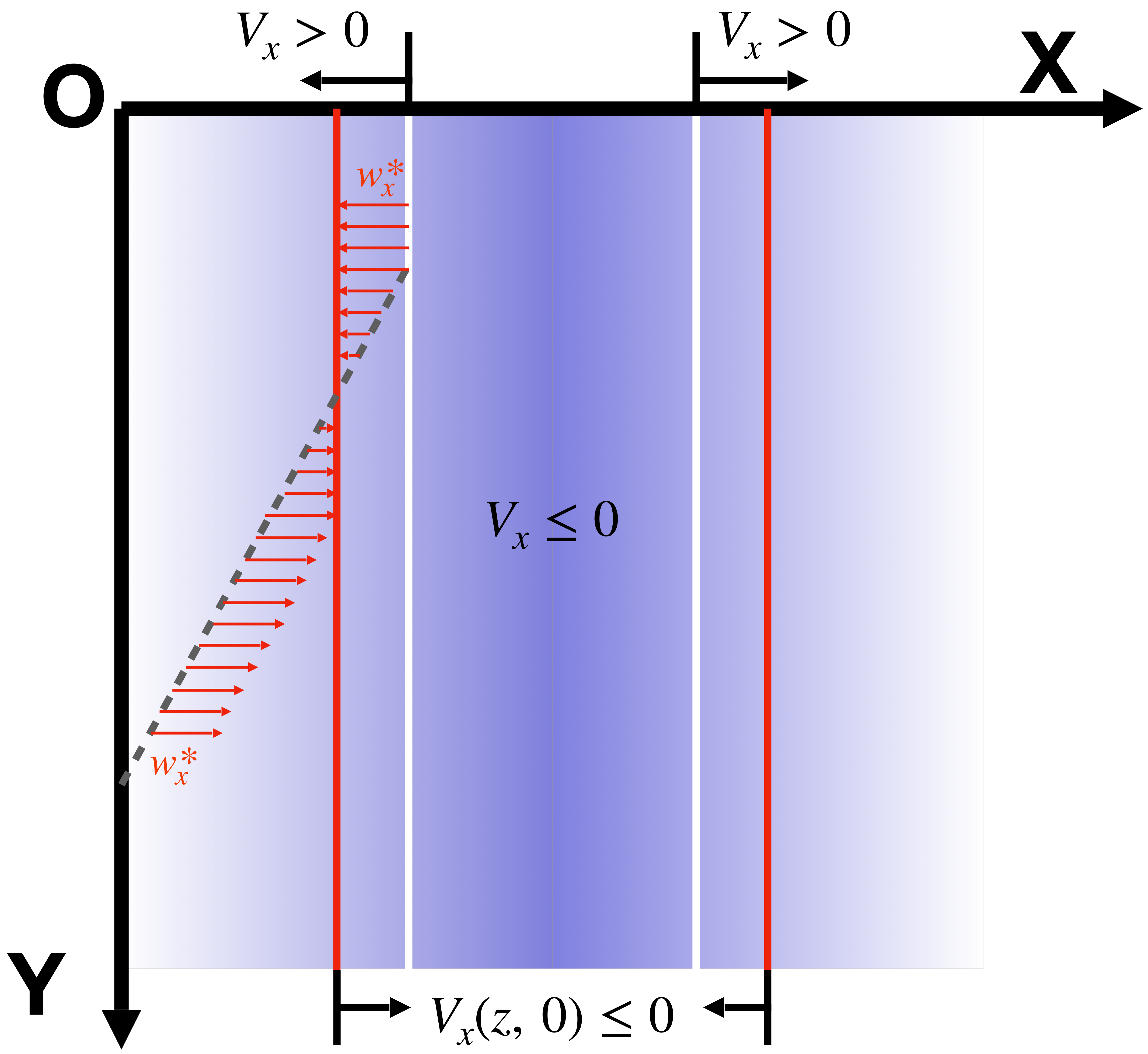
Avoiding zero sub-level set

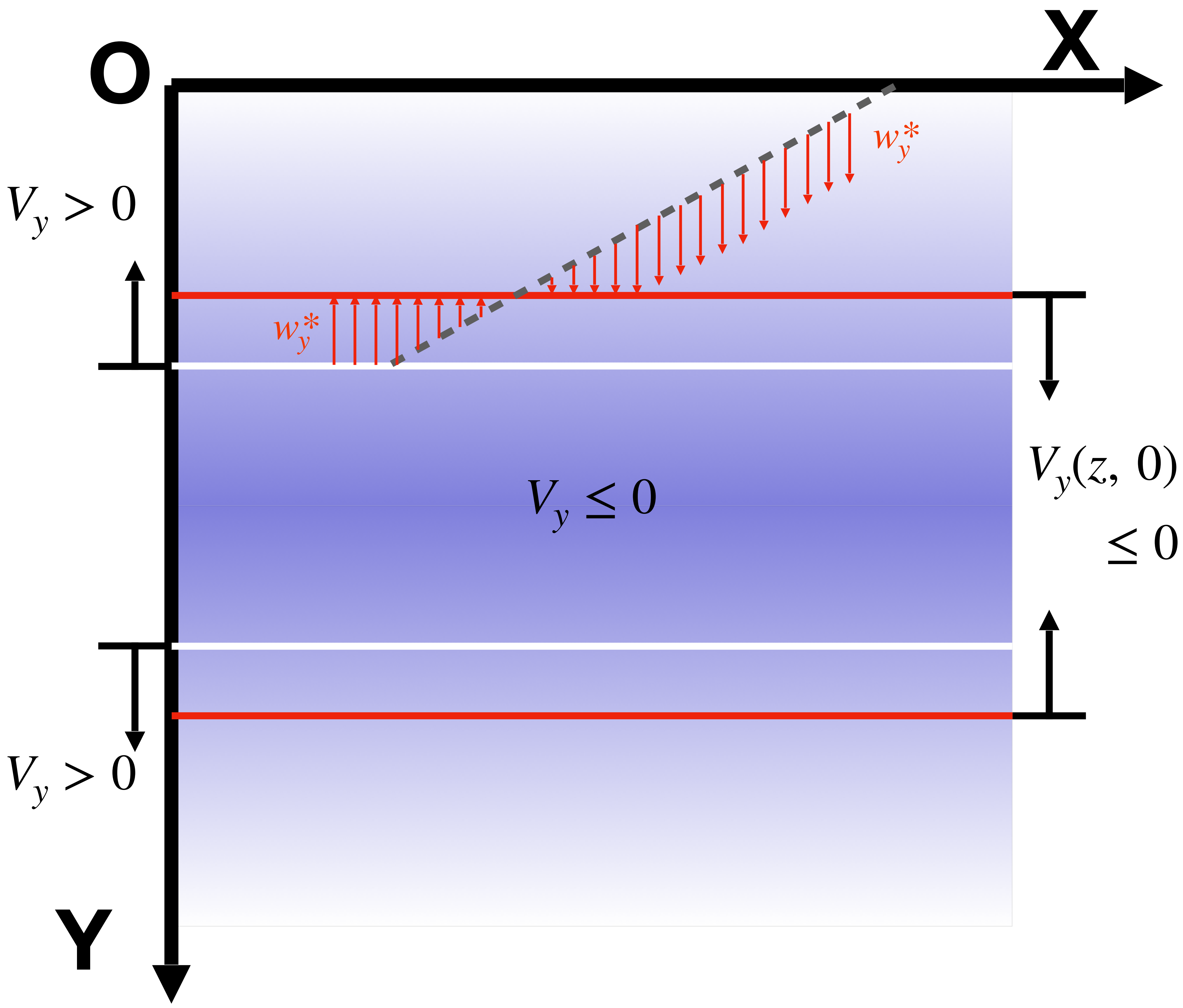




Full-dimensional sub-value functions

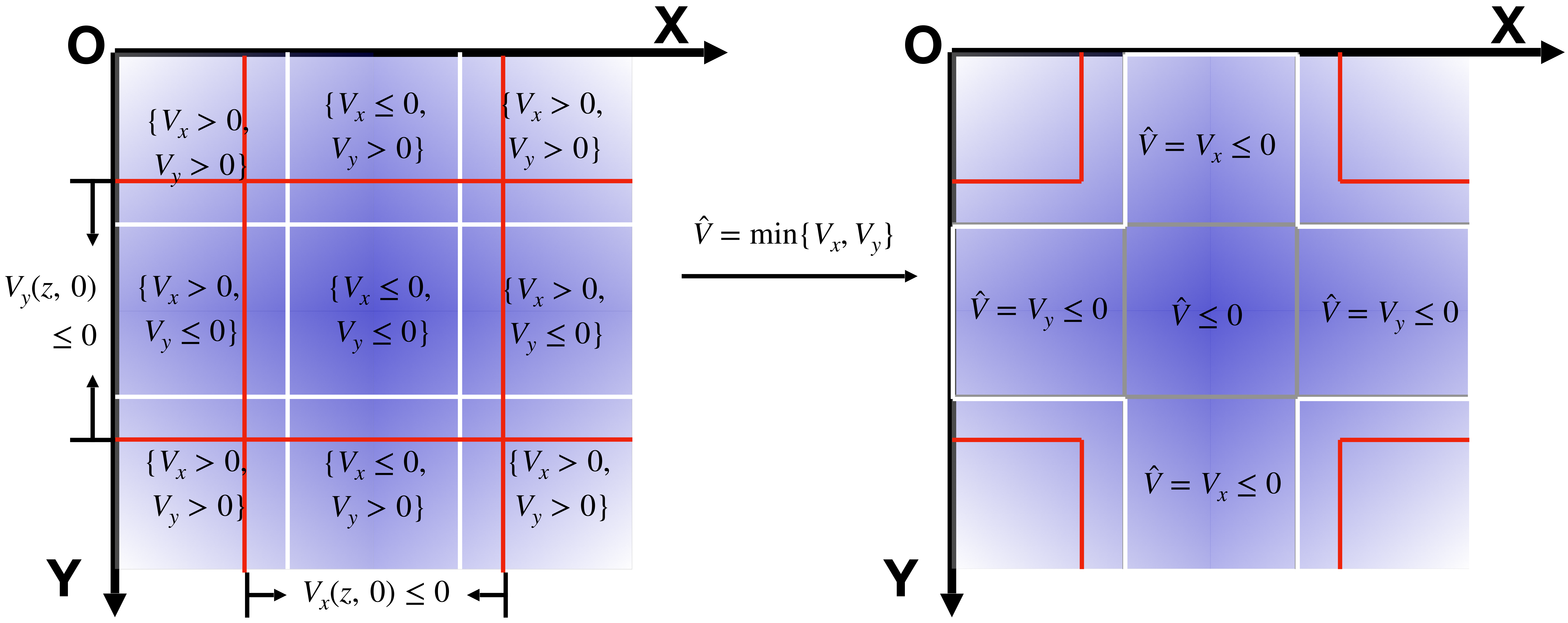
Avoiding zero sub-level set (Possible Controls)





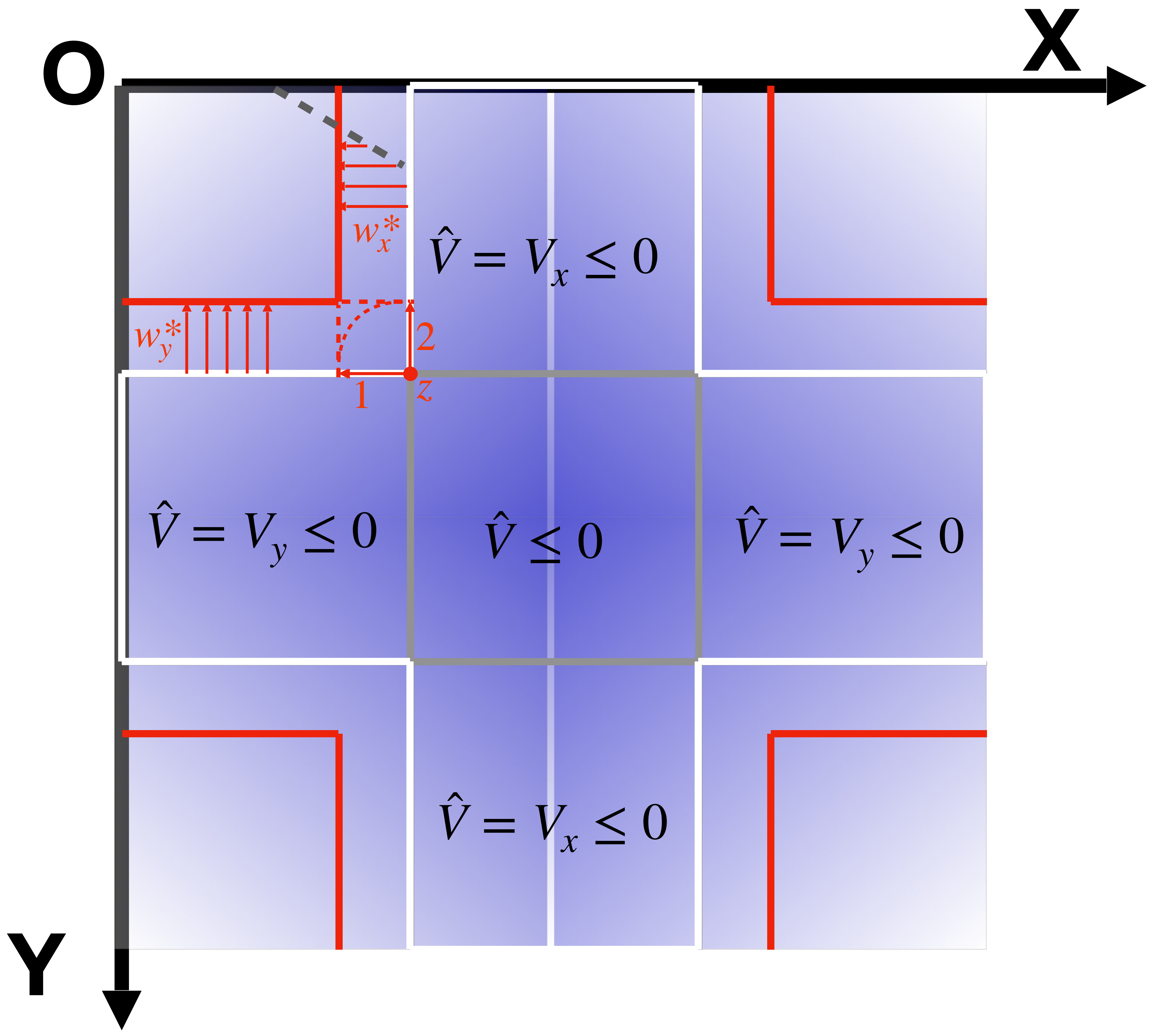
Full-dimensional approximated Value Function

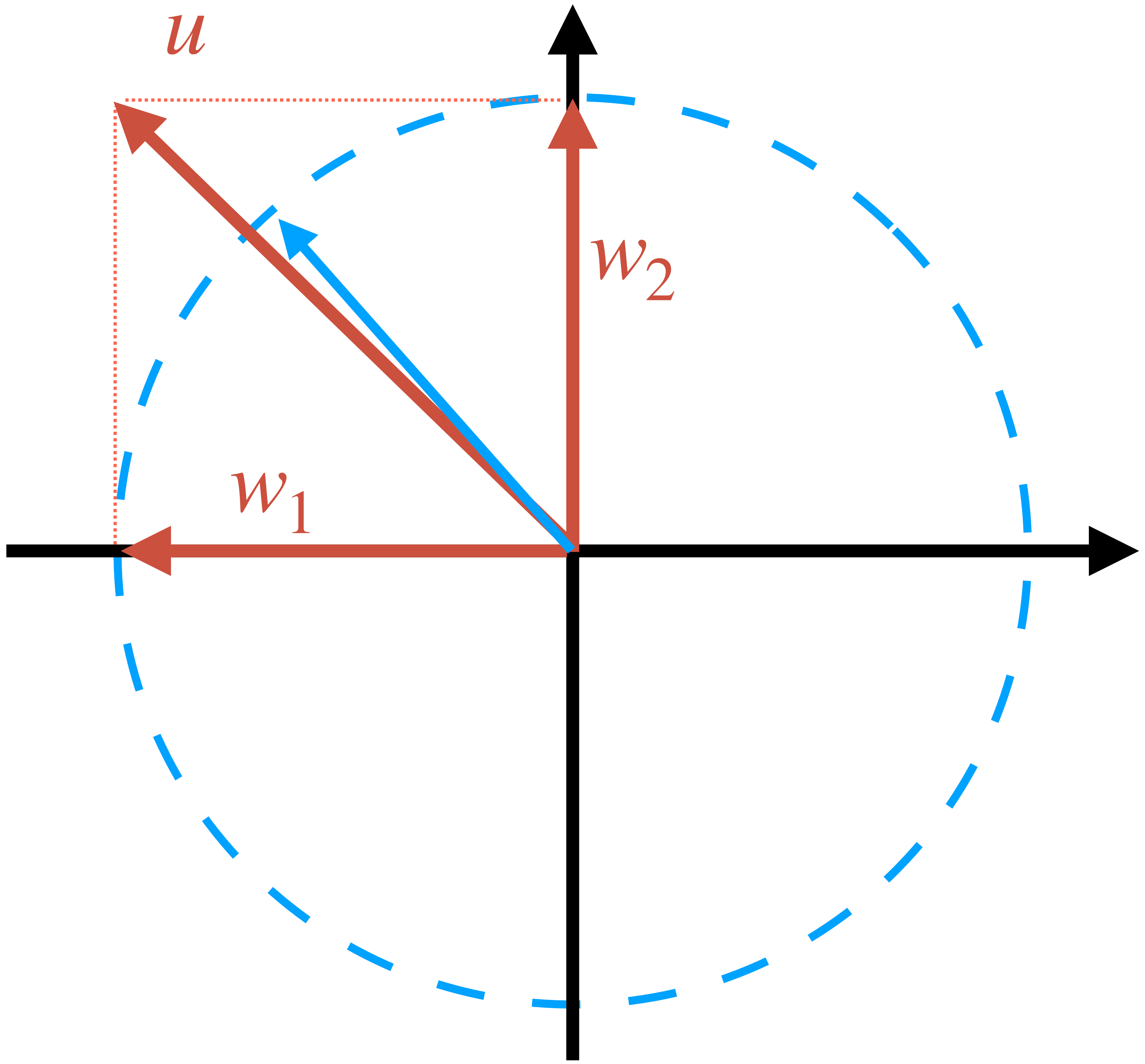
Avoiding the union zero sub-level set



Leaking Corner Issue

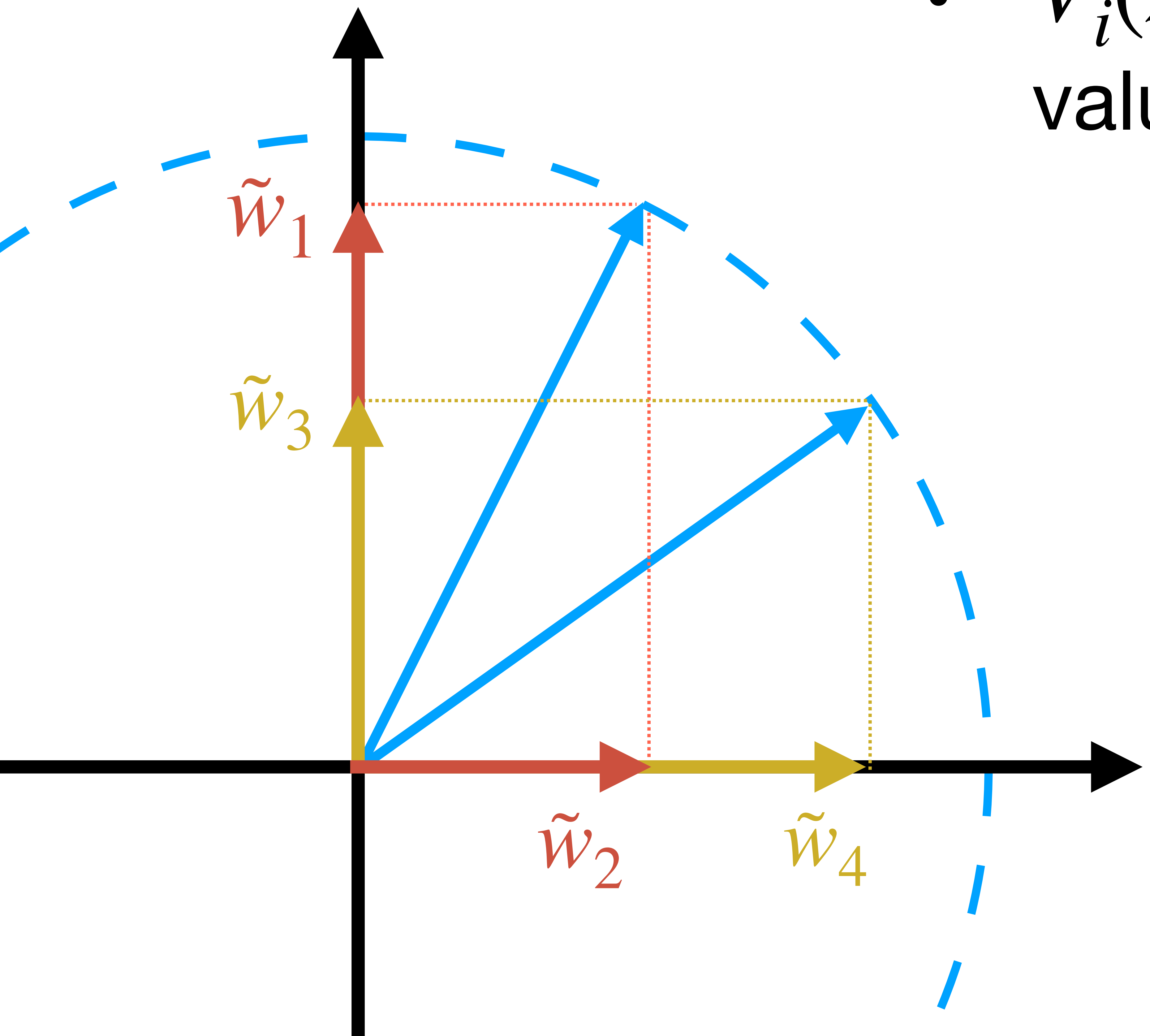
Unrealistic Controls





Allowable Control

- $(\tilde{w}_1, \tilde{w}_2)$: a pair of low-dimensional control signals that satisfy all the control constraint.
- $\tilde{V}_i(z, t)$: The corresponding full-dimensional sub-value functions



$\tilde{w}_1$ and $\tilde{w}_2$ are a pair of allowable controls;
$\tilde{w}_3$ and $\tilde{w}_4$ are a pair of allowable controls.

Allowable Control

its relation to the leaking corner

The states not suffereing from the “leaking corner issue”:

- Intersection for liveness problem: $\max\{\tilde{V}_{R,1}(z,t), \tilde{V}_{R,2}(z,t)\} = \hat{V}_R(z,t)$,
- Union for safety problem: $\min\{\tilde{V}_{A,1}(z,t), \tilde{V}_{A,2}(z,t)\} = \hat{V}_A(z,t)$.

Lemma 2 from our paper

Threshold Strategy

1. We can find the set of leaking corners $\mathscr{L}(t)$ by comparing the (full-dimensional) sub-value functions:

$$\mathscr{L}(t) = \{z : |V_1 - V_2| < \Delta\}.$$

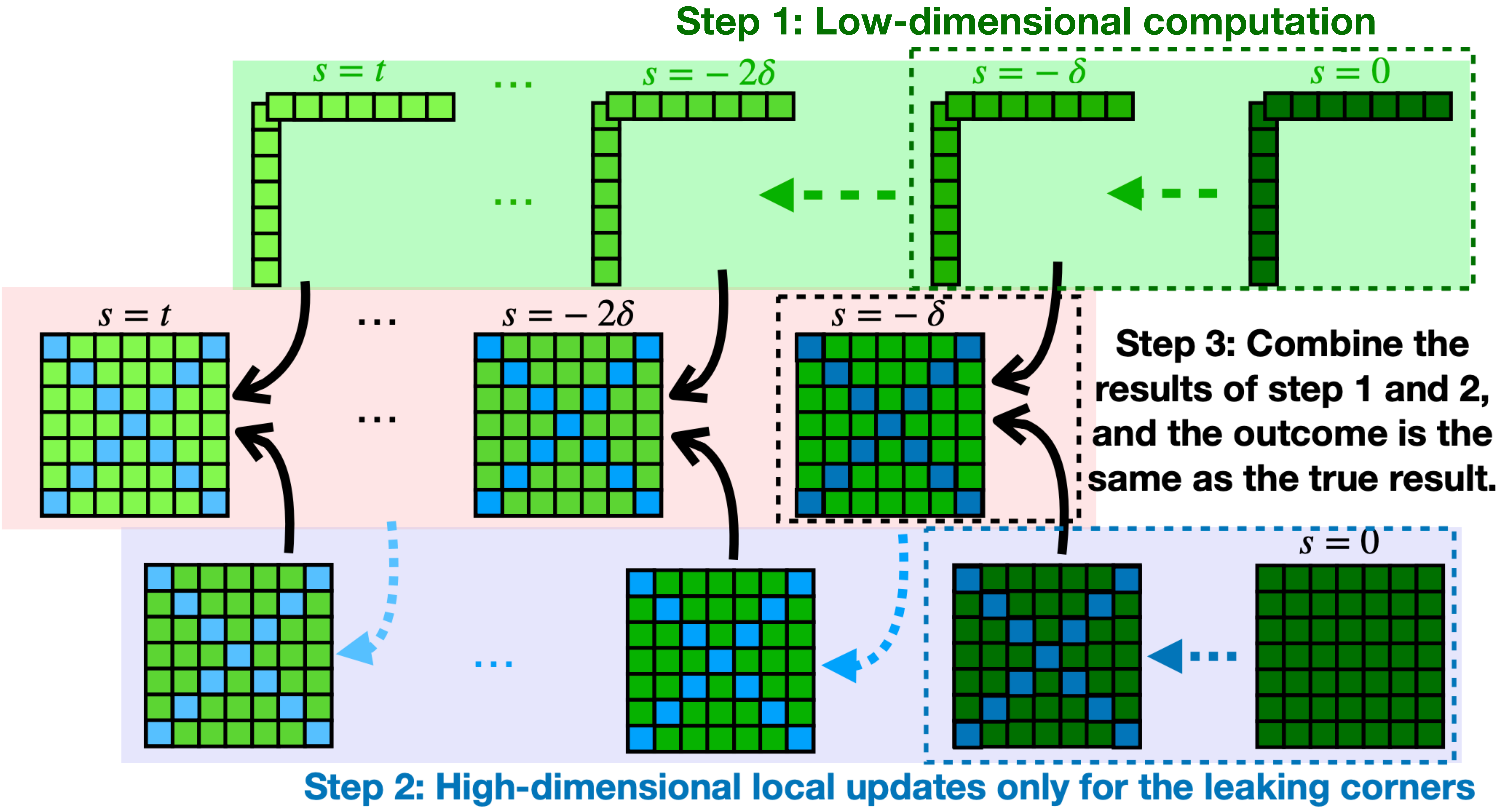
Refer to Theorem 1 from our paper to see how to find Δ value.

2. A local updating method starting from the states:

$$\{z : V_1(z, t) = V_2(z, t)\}$$

and will then cover all the leaking corners.

Updating Method



Algorithm 1: Local updating procedure

Data: $\hat{V}(\cdot,\cdot), \hat{\mathcal{L}}(\cdot), Z, t_{\text{list}} = [t, t+\delta..., 0]$
Result: $\check{V}(\cdot,\cdot)$

```
s ← 0;                                   ▷ Backward Computation
V̌(·,0) ← V̂(·,0);
Frontier ← nextFrontier ← visited ← {};
while s > t do
    for z ∈ Z do
        if z ∈ L̂(s) then
            updateValue(z, s, δ, Frontier)
        else
            V̌(z, s − δ) ← V̂(z, s − δ);
        end
    end
    visited ← L̂(s)
    Frontier ← Frontier \ visited
    while Frontier ≠ Ø do
        for z in Frontier do
            updateValue(z, s, δ, nextFrontier)
        end
        visited ← visited ∪ Frontier
        Frontier ← nextFrontier \ visited
        nextFrontier ← {}
    end
    s ← s − δ;
end
def updateValue(z, s, δ, Frontier):
    V̌(z, s − δ) ← HJ Update(V̌(z, s))   ▷ Equation 5
    if V̌(z, s − δ) ≠ V̂(z, s − δ) then
        Frontier ← Frontier ∪ neighbor(z)
    end
```

Only compute the potential leaking corners in full-dimensional space

Update the Frontier while it is non-empty

Include the neighbors as Frontier if the state is the leaking corner

The procedure makes sure that all the leaking corners are covered with the updating procedure.

Results

2D Single Integrator

TABLE I: 2D Accuracy Comparison for One Step

Metric	Before	After
Number of grid points with different values from the ground truth	200	0
Average absolute difference from ground truth	1.2×10^{-4}	9.51×10^{-18}
Maximum absolute difference from ground truth	2×10^{-2}	2.22×10^{-16}

TABLE III: 2D Accuracy Comparison for 10 Steps

Metric	Before	After
Number of states with different values from the ground truth	1344	0
Average absolute difference from ground truth	2.5×10^{-3}	1×10^{-9}
Maximum absolute difference from ground truth	7.39×10^{-2}	2.44×10^{-8}

TABLE II: 2D Time Comparison for One Step

Process	Time (seconds)
Direct computation	3.3×10^{-2}
SCSD computation + HJ local update computation	$7 \times 10^{-4} + 1.3 \times 10^{-3} = 2.0 \times 10^{-3}$

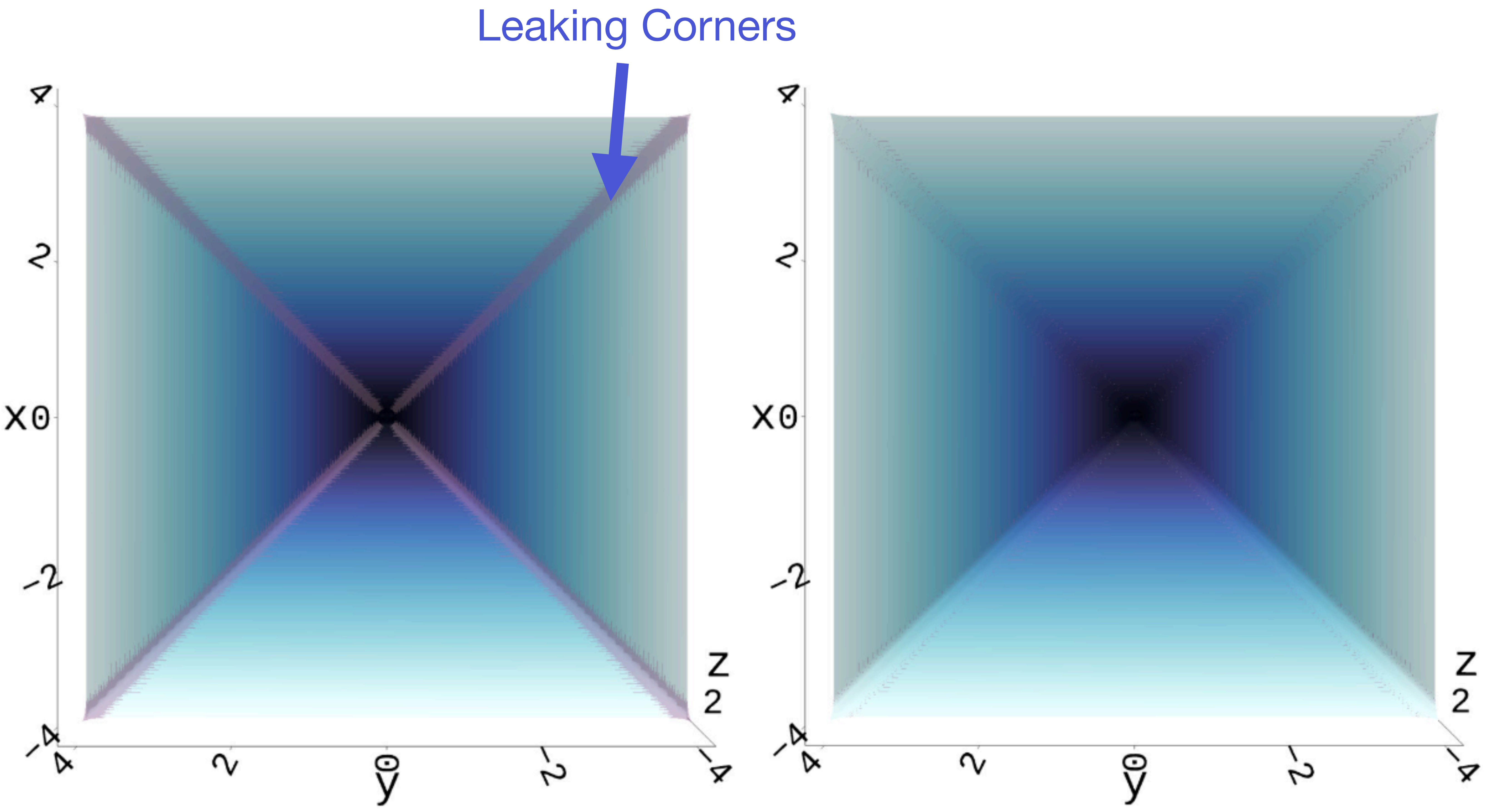


TABLE IV: 2D Time Comparison for 10 Steps

Process	Time (seconds)
Direct computation	3.36×10^{-1}
SCSD computation + HJ local update computation	$4.72 \times 10^{-3} + 1.61 \times 10^{-1} = 1.65 \times 10^{-1}$

- Successfully recompute all the leaking corners
- Computationally efficient compare to the direct computation

Results

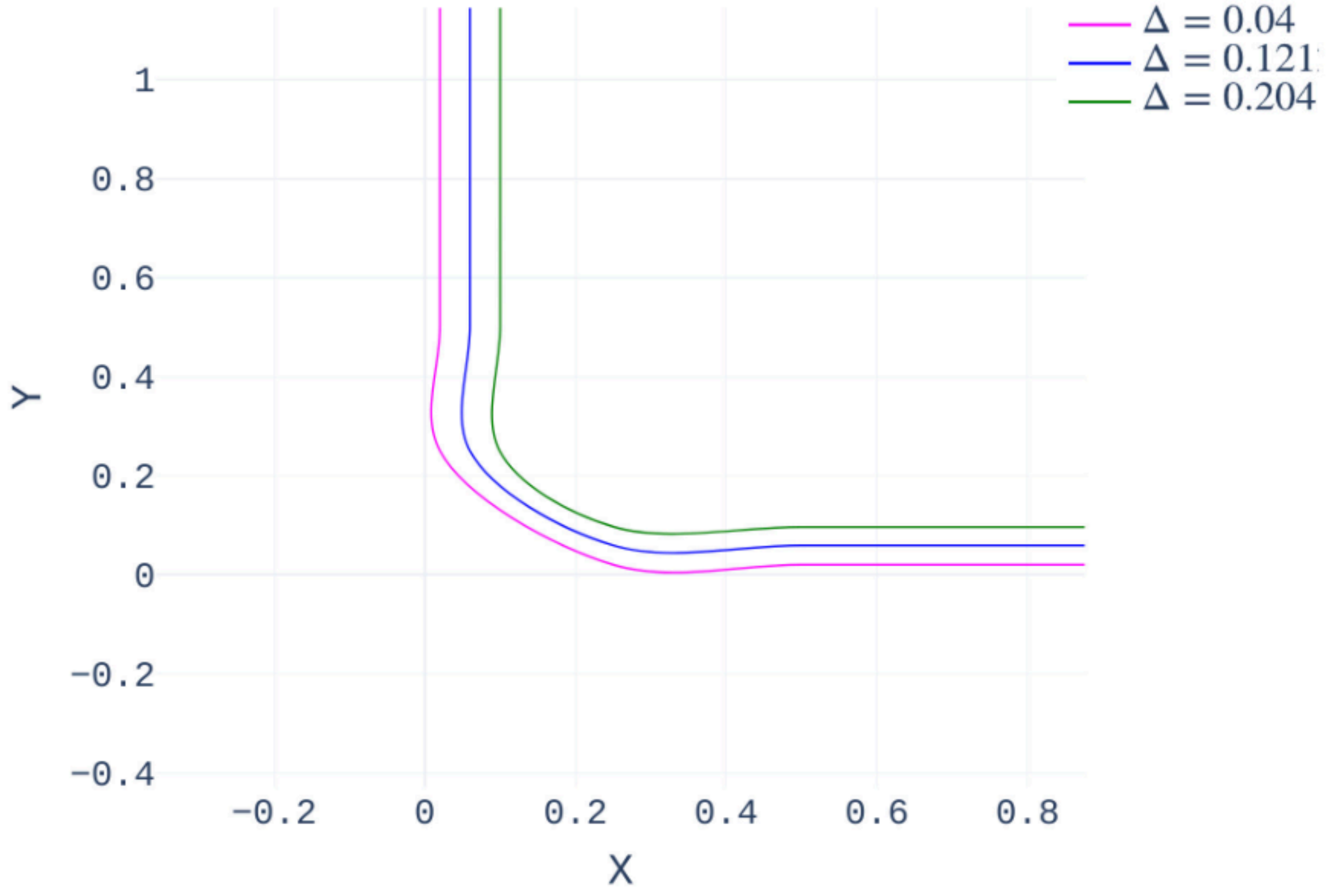
6D Planar Quadrotor

TABLE V: 6D Accuracy Comparison for One Step

Metric	Before	After
Number of grid points with different values from the ground truth	3.89×10^6	0
Average absolute difference from ground truth	6.97×10^{-4}	0.0
Maximum absolute difference from ground truth	4×10^{-2}	0.0

TABLE VI: Computation Time and Delta Value for ts

t (s)	Δ	Decomposition Time + Local Updating Time (seconds)
-0.02	0.04	2.447 + 47.1078 = 49.5548
-0.06	0.1212	6.769 + 157.3528 = 164.1218
-0.1	0.204	11.8038 + 250.7859 = 262.5897



Conclusion

1. Propose a threshold-based method to detect the leaking corners.
2. Introduce a local updating method that ensures accuracy while maintaining computational efficiency.
3. Validate the method with 2D Single Integrator system and 6D Planar Quadrotor system.

Future Works

1. Implement our method to other techniques involving the combination of sub-value functions.
2. Parallelize the local updating procedure for faster computation
3. Explore machine learning or other methods for new value updating methods.

Thank you!

Project webpage

Contact: Chong He
Email: chong_he@sfu.ca