Continuous-Time VeRecycle

Reclaiming Reach-Avoid Guarantees from Neural Certificates for Stochastic Dynamical Systems after Localised Changes

Sabina-Mălina Grădinariu[1]
Supervisor: Dr. Anna Lukina[1]
Daily supervisor: Sterre Lutz[1]
Evaluation committee: Dr. Anna Lukina[1]**, Sterre Lutz**[1]**, Dr. Luca Laurenti**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

June 3, 2026

Contents

Abstract

Safety-critical autonomous systems, such as robots or drones, often depend on formal guarantees. These show that they will reach a desired target while avoiding unsafe states. These guarantees can be provided by a certificate: a mathematical proof object that summarises the system's behaviour without requiring all possible trajectories to be simulated. However, once the system is deployed, its dynamics may change locally, for example, because of altered friction, disturbances, or unexpected environmental effects. Recomputing a new certificate from scratch can be expensive, especially when the certificate is represented by a neural network.

This thesis studies whether an existing certificate can still be reused when such a localised change occurs in a continuous-time stochastic system. It extends the VeRecycle reclaiming principle, originally developed for discrete-time stochastic dynamical systems, to continuous-time stochastic differential equations. VeRecycle provides a mathematical framework for determining how much of an existing reach-avoid guarantee can still be recovered after the dynamics change within a known region of the state space. This extension is important because many physical control systems evolve continuously, while the existing VeRecycle theory is formulated only for discrete-time models.

The main contribution is a continuous-time reclaiming rule for reach-avoid certificates. The rule shows that, for a fixed certified tuple $(V, \alpha, \beta, \zeta)$, the maximal guarantee that can be recovered by reusing the certificate V depends on the changed region only through the minimum certificate value over that region. In other words, the changed region affects the reclaimed guarantee only through the lowest value that the original certificate reaches on that region.

The framework is implemented using neural reach-avoid certificates and interval-bound propagation, and is evaluated on a stochastic inverted pendulum benchmark. The performed experiments show that the recovered guarantees depend on the local certificate minimum, and can be computed considerably faster than full re-certification, while avoiding failure modes caused by naive discretise-then-reclaim approaches.

1 Introduction

Let us consider a stochastic inverted pendulum controlled by a fixed feedback policy. The system state consists of the pendulum's angular velocity and angle, and the goal is to reach a desired target region while avoiding unsafe states. Assuming that a neural reach-avoid certificate has already been computed for this system, it gives a formal lower bound on the probability of reaching the target safely. If the dynamics later change only in a known region of the state space, for example due to altered friction or disturbance, it is natural to ask whether the existing certificate can still be reused instead of recomputing a new one from scratch. This is the central problem studied in this thesis.

Safety-critical autonomous systems are rarely deployed in static environments. Even with an initially certified model, localised changes in dynamics may invalidate the original guarantee, because the certificate conditions are verified for the original system dynamics. Once the dynamics change inside a region of the state space, the original expected-decrease or generator condition may no longer hold there. Recomputing a formal guarantee from scratch can be costly, especially when certification relies on neural proof certificates (Lutz et al., 2025).

To the best of our knowledge, the first framework to formally reclaim guarantees after localised changes in stochastic dynamical systems is VeRecycle, introduced by Lutz et al. (2025). VeRecycle assumes discrete-time stochastic dynamics and builds on the probabilistic supermartingale certificates introduced by Zikelic et al. (2023). Rather than re-certifying the full system after a localised change, VeRecycle shows that one can reuse an existing certificate and recover a valid reach-avoid guarantee. However, many physical control systems evolve continuously over time. Although control and verification are frequently implemented using discrete-time abstractions, the underlying physical motion of robots, vehicles, and drones is in reality continuous (Neustroev et al., 2025). This creates an important limitation: existing VeRecycle theory cannot directly reason about systems whose evolution is fundamentally continuous, and consequently cannot reason about the continuous-time model underlying real-world control. A natural next step is therefore to broaden the VeRecycle reclaiming principle to continuous-time stochastic dynamical systems.

Recent work on probabilistic verification of learning-enabled stochastic systems has recently relied more on neural certificate techniques that enforce Lyapunov, or supermartingale-type decrease conditions. This line of research can be traced back to termination arguments for probabilistic programs based on ranking supermartingales, which was later combined with neural learning and formal verification (Abate et al., 2021b; Giacobbe et al., 2022). In discrete time, these methods have enabled quantitative reach-avoid guarantees for stochastic dynamical systems under neural control (Abate et al., 2024; Ansaripour et al., 2023; Badings et al., 2024; Chatterjee et al., 2023; Mathiesen et al., 2023; Zikelic et al., 2023). These works show that neural certificates can provide quantitative reach-avoid guarantees for stochastic systems, but they do not address whether such guarantees can be reused after localised model changes in continuous time.

In parallel, the field of neural certificates has recently progressed toward continuous-time reasoning. The first formulation of neural supermartingale certificates for continuous-time stochastic differential equations was introduced by Neustroev et al. (2025). Their work develops a continuous-time analogue of probabilistic Lyapunov and barrier certificates (Kalman and Bertram, 1960; Prajna et al., 2004). This closes a conceptual gap between probabilistic verification and continuous-time stochastic control. However, the method does not address the question of what happens when the dynamics alter suddenly in a specific region of the state space, nor how to reuse an existing neural certificate without recomputing it entirely.

This thesis aims to connect these two lines of research. Our goal is to develop the first continuous-time formulation of VeRecycle and to evaluate whether certificate reuse retains its computational benefit in this setting. We aim to unify the need for localised certification reuse with the right mathematical tools for continuous processes. More precisely, we investigate whether a previously certified continuous-time reach-avoid certificate can still provide a universally valid guarantee following an arbitrary localised change in the dynamics, without retraining the certificate itself. We also compare the computational cost of reclaiming with full re-certification and study failure modes of naive discretise-then-reclaim workflows. The outcome is a framework that extends probabilistic certification beyond discrete-time abstractions and addresses the continuous-time dynamics that autonomous systems operate in.

1.1 Motivation

Why continuous time matters. Discrete-time models are convenient for simulation and learning, but they are often abstractions of continuous-time dynamics. Transitioning among discrete states at fixed time steps ignores the fact that the system develops continuously and can encounter unsafe events between sampling instants. Moreover, the behaviour of a continuous-time stochastic system is controlled by its drift and diffusion functions, not by transition kernels, which directly encode how uncertainties act at every instant. Relying exclusively on discrete-time certificates, thus, means that such formal guarantees remain tied to a discrete abstraction instead of the physical system itself (Neustroev et al., 2025).

Why formal guarantees matter. This matters especially in safety-sensitive settings where certification is needed not only to assess how a system behaves in simulations or experiments, but also to provide assurance before deployment and during operation. Simulations can suggest that a controller behaves safely on many sampled trajectories, but they can only explore finitely many behaviours of a stochastic system. They therefore cannot rule out all safety-critical failures. Certificate-based verification is valuable precisely because it provides mathematically strict guarantees that hold uniformly over all admissible behaviours of the model. For systems functioning under uncertainty, such guarantees are particularly important, since safety violations may

appear from stochastic effects that are unlikely to be observed in simulation yet remain unacceptable in deployment (Baier and Katoen, 2008; Clarke et al., 2018).

Why localised change matters. In many real-life situations, safety must be ensured even when dynamics change unexpectedly. A robot walking on a wet surface, an industrial arm experiencing friction, or an autonomous drone entering a region with unknown wind forces are all examples of localised differences in the dynamics. In such cases, the nominal model may remain accurate almost everywhere, while failing only in a restricted subset of the state space. Nevertheless, even a localised deviation may invalidate a previously certified guarantee, because the certificate conditions are verified for the original dynamics. Once the dynamics change inside the affected region, the expected-decrease or generator condition may no longer hold there, so the original certificate no longer automatically proves the same guarantee (Lutz et al., 2025).

Why reuse matters. A natural response would be to recompute a new certificate for the modified system. However, this is costly, especially in learning-based verification pipelines where certificates are represented using neural networks and must be retrained and reverified. This makes certificate reuse highly desirable. VeRecycle was explicitly designed to handle such cases in discrete time by reclaiming a valid reach-avoid guarantee without retraining a neural certificate (Lutz et al., 2025). Developing the continuous-time analogue is valuable because it extends this ability to the systems where it is most needed. Continuous-time VeRecycle allows us to combine the strengths of continuous-time probabilistic verification with the flexibility of using certificates learned before the model changed. This is relevant for safety-critical systems, where model updates cannot be performed instantly, and guarantees must remain valid under uncertainty.

Why the extension is nontrivial. Apart from its practical relevance, the problem is also conceptually nontrivial. In the continuous-time setting considered in this thesis, certification is expressed through generator-based decrease conditions, whereas the discrete-time VeRecycle setting relies on one-step stochastic transitions and their associated certificate conditions (Neustroev et al., 2025; Lutz et al., 2025). It is thus not immediate that the reclaiming principle from discrete time should carry over in a similarly clean form. A main theoretical contribution of this thesis is to prove that, in spite of these differences, the maximal reclaimed guarantee certifiable by reusing the fixed certificate can still be characterised through a single local quantity: the infimum of the certificate over the changed region.

Research question. This thesis studies the following question: *Can the VeRecycle reclaiming principle be soundly extended to continuous-time stochastic dynamical systems, and does the resulting framework both characterise reclaimed guarantees through the certificate infimum over the modified region and preserve the computational benefit of certificate reuse?*

We examine this main question through two sub-questions:

RQ1 Can the VeRecycle reclaiming principle be soundly extended to continuous-time stochastic systems using generator-based reach–avoid certificates, and can the maximal reclaimed guarantee be characterised solely through the certificate infimum

$$m = \inf_{x \in X_q} V(x)$$

over the modified region?

RQ2 Does continuous-time VeRecycle provide a computational benefit over full re-certification, and which concrete experiments expose failure modes of naive discretise-then-reclaim workflows?

Objective. This thesis presents the first continuous-time formulation of VeRecycle. Specifically, we investigate whether the reclaimed reach-avoid guarantee certifiable by reusing a fixed continuous-time certificate under localised model change can be characterised solely in terms of the certificate's infimum over the modified region. In addition, we evaluate whether this reuse leads to computational savings compared with full re-certification, and we describe experiments that expose limitations of naive discretise-then-reclaim workflows. Our goal is to define a precise conceptual framework, prove its soundness, and assess its behaviour through implementation and experiments.

1.2 Contributions

The main contributions of this thesis are the following:

- **Sound continuous-time extension of VeRecycle (RQ1).** We prove that the VeRecycle reclaiming principle can be applied soundly to continuous-time stochastic systems modelled by stochastic differential equations and certified using generator-based reach–avoid certificates. The framework keeps the original certificate fixed after a localised model change and characterises the maximal reclaimed reach–avoid guarantee through the certificate infimum over the modified region.
- **Experimental assessment and comparison (RQ2).** We evaluate the framework on a stochastic inverted pendulum benchmark. The experiments show that the computed reclaimed guarantees are consistent with the theoretical reclaiming rule, measure runtime savings over full re-certification, and expose failure modes of naive discretise-then-reclaim workflows.

1.3 Thesis structure

The chapters ahead are structured as follows. Chapter 2 introduces the background on certificate-based verification, stochastic dynamical systems, and the methods on which this work builds. Chapter 3 presents the system model, the reach-avoid objective, and the formulation of localised model changes. Chapter 4 develops the continuous-time VeRecycle framework and presents the main reclaiming result. Chapter 5 evaluates the framework on a stochastic inverted pendulum benchmark. The thesis concludes with a discussion of the results, limitations, wider impact, and directions for future work.

2 Background

This chapter lays out the core concepts and technical foundations used throughout the thesis. Its purpose is not only to review related literature, but also to build the chain of ideas that leads to the continuous-time VeRecycle result, which is proved later. The thesis sits at the intersection of two lines of work: probabilistic reach–avoid certification for continuous-time stochastic systems and certificate reuse under localised model change.

The chapter is organised as follows. First, it explains why formal verification is needed for stochastic dynamical systems and how certificate functions provide a tractable alternative to reasoning about trajectories one by one. It then presents the main certificate families relevant to this thesis: Lyapunov functions, barrier certificates, and supermartingale certificates. Next, it discusses probabilistic reach-avoid certificates in discrete time, because they provide the clearest intuition for how reachability and safety are combined through level-set reasoning. The chapter then turns to continuous-time stochastic systems and explains the infinitesimal-generator viewpoint that underlies the framework of Neustroev et al. (2025). Finally, it discusses localised model change and the discrete-time VeRecycle framework of Lutz et al. (2025), which motivates the reclaiming problem studied in this thesis.

Throughout this chapter, we use the notation that will be formalised in Chapter 3. In particular, X denotes the state space, X_0 the initial set, $X_\star$ the target set, $X_\oslash$ the unsafe set, and X_q the region where the dynamics may change. We write π for the fixed feedback policy, f_π and g_π for the original closed-loop drift and diffusion, G_π for the corresponding infinitesimal generator, and V for a certificate function. When reviewing prior work, we translate the notation of those papers into this notation whenever this improves readability.

2.1 Formal verification of stochastic dynamical systems

Formal verification aims to establish mathematically correct guarantees about the behaviour of a system under all admissible executions, rather than only under a finite set of simulated scenarios (Clarke et al., 2018; Baier and Katoen, 2008; Tabuada, 2009). This distinction is especially important for safety-critical dynamical systems operating under uncertainty. In such settings, a controller may appear reliable in simulation, while still failing on rare but safety-critical trajectories, because any simulation campaign can only cover finitely many behaviours.

A central difficulty is that stochastic dynamical systems generate infinitely many possible trajectories. Even if the model and controller are fixed, random disturbances may lead to different paths from the same initial state. Some paths may reach the target, while others may enter the unsafe set.

Continuous-time stochastic model. Following Neustroev et al. (2025), we model such systems using a stochastic differential equation (SDE)

$$dX_t = f(t, X_t, u_t)\, dt + g(t, X_t, u_t)\, dW_t.$$

Here, $X_t \in \mathbb{R}^\ell$ is the state at time t, $u_t \in U$ is the control input, f is the drift, g is the diffusion, and W_t is a Wiener process. The drift describes the deterministic part of the motion, while the diffusion describes how stochastic noise enters the system.

Closed-loop dynamics. In this thesis, the control input is chosen by a fixed feedback policy,

$$u_t = \pi(X_t),$$

meaning that the current state determines the applied control action. Substituting this policy into the SDE gives the closed-loop dynamics

$$dX_t = f_\pi(X_t)\, dt + g_\pi(X_t)\, dW_t, \tag{1}$$

where

$$f_\pi(x) := f(x, \pi(x)), \qquad g_\pi(x) := g(x, \pi(x)).$$

Thus, f_π is the drift after the policy has been fixed, and g_π is the corresponding diffusion. The goal of verification is to prove statements about all sample paths of (1), or about the probability with which they satisfy a given specification.

Running example. ***Stochastic inverted pendulum.*** *As a running example, consider a stochastic inverted pendulum controlled by a fixed feedback policy. The state is*

$$x = (\omega, \theta),$$

where ω is the angular velocity in rad/s and θ is the pendulum angle in radians. In the notation of Neustroev et al. (2025), the angular velocity coordinate is denoted by $\phi = \dot{\theta}$; here we write ω to preserve consistency with the rest of the thesis.

The benchmark is an instance of the closed-loop SDE in (1)*. In first-order form, its controlled stochastic dynamics can be written as*

$$d\omega_t = \left(\frac{g}{L} \sin(\theta_t) + \frac{M\pi(\omega_t, \theta_t) - b\omega_t}{m_\mathrm{p} L^2} \right) dt + \sigma\, dW_t,$$
$$d\theta_t = \omega_t\, dt.$$

The first equation describes the angular-velocity dynamics under gravity, control, damping, and stochastic disturbance. The second equation states that the angle evolves according to the angular velocity. Here, g, L, M, m_p, b, and σ denote the physical and noise parameters of the benchmark; their concrete values are given in the evaluation setup.

Because the system is stochastic, two trajectories starting from the same state may evolve differently. The verification question is therefore not whether one simulated trajectory succeeds, but whether the stochastic system satisfies the desired property with sufficiently high probability. The benchmark is visualised in Figure 1.

An important class of specifications is reach–avoid: starting from an initial set X_0, the system has to reach a target set $X_\star$ before entering an unsafe set $X_\oslash$. In the notation used later in this thesis, this is captured by the stopping time

$$\tau := \inf\{t \geq 0 \mid X_t \in X_\star \cup X_\oslash\}, \tag{2}$$

and by the probability

$$\mathbb{P}_{x_0}(X_\tau \in X_\star). \tag{3}$$

The challenge is to obtain a sound lower bound on (3) without enumerating or approximating the full reachable set of the stochastic process.

Running example. ***Stochastic inverted pendulum, continued.*** *For the pendulum benchmark, the initial set is*

$$X_0 = [-1, 1] \times [3\pi/4, 5\pi/4],$$

which represents states with small angular velocity and angle close to π. The target set is

$$X_\star = [-4, 4] \times [-\pi/2, \pi/2],$$

which represents successful control: the pendulum has moderate angular velocity and its angle is close enough to the desired controlled state $(\omega, \theta) = (0, 0)$. The unsafe set is

$$X_\oslash = ([-20, -10] \times [-2\pi, -3\pi/2]) \cup ([10, 20] \times [3\pi/2, 2\pi]).$$

Thus, unsafe states correspond to high-speed, large-angle configurations. The reach–avoid probability measures how likely it is that the pendulum reaches $X_\star$ before entering $X_\oslash$. States outside $X_\oslash$ are not automatically safe; they are simply not part of the explicitly unsafe set.

2.2 Certificate-based reasoning

Why certificates are useful. A standard way to avoid explicit trajectory enumeration is to use a certificate function. In deterministic control, this idea appears in Lyapunov-style reasoning (Kalman and Bertram, 1960; Khalil, 2002); for safety and stochastic verification, it appears in barrier and supermartingale-style certificates (Prajna and Jadbabaie, 2004; Prajna et al., 2007; Neustroev et al., 2025).

Conceptually, a certificate assigns a value $V(x)$ to each state x. In a reach–avoid setting, lower certificate values indicate states for which reaching the target before the unsafe set is certified more favourably, while higher values indicate states that are closer to the unsafe certificate level. The certificate value should thus not be interpreted as the true reach–avoid probability, but as a quantity from which a probability bound can be derived. Verification is reduced to proving inequalities involving V, such as decrease conditions, boundary conditions, or separation conditions between initial, target, and unsafe regions.

Definition 1 (Certificate function)**.** *A certificate function is a scalar function $V : X \to \mathbb{R}$ whose properties imply a behavioural guarantee for the underlying system. Depending on the verification objective, these properties may take the form of positivity conditions, boundary-separation conditions, or decrease conditions along the dynamics.*

Running example. ***Stochastic inverted pendulum, continued.*** *For the pendulum, a certificate V can be imagined as a surface over the (ω, θ)-plane. A point on this plane is a physical state of the system, and the height of the surface is the certificate value. The goal of certification is not to simulate every possible noisy trajectory, but to prove that this surface has the right shape: low on allowed initial states, high on unsafe states, and decreasing in expectation before the target is reached.*

Lyapunov functions. The historical origin of certificate-based reasoning lies in Lyapunov stability theory. Consider a deterministic system

$$\dot{x} = f(x), \tag{4}$$

with an equilibrium point $x^\star$. A Lyapunov function is a scalar function V that acts as an energy-like measure of deviation from the equilibrium. It is typically chosen such that

$$V(x^\star) = 0, \qquad V(x) > 0 \quad \text{for all } x \neq x^\star.$$

Thus, the equilibrium has the lowest possible value, while all other states have positive values.

The key idea is that V should decrease along system trajectories. Since V is minimal at the equilibrium, decreasing values of V force trajectories into lower sublevel sets of V, which are around the equilibrium. Under the standard Lyapunov conditions, this implies that the system remains close to the equilibrium, and with a strict decrease condition, it can establish convergence toward it.

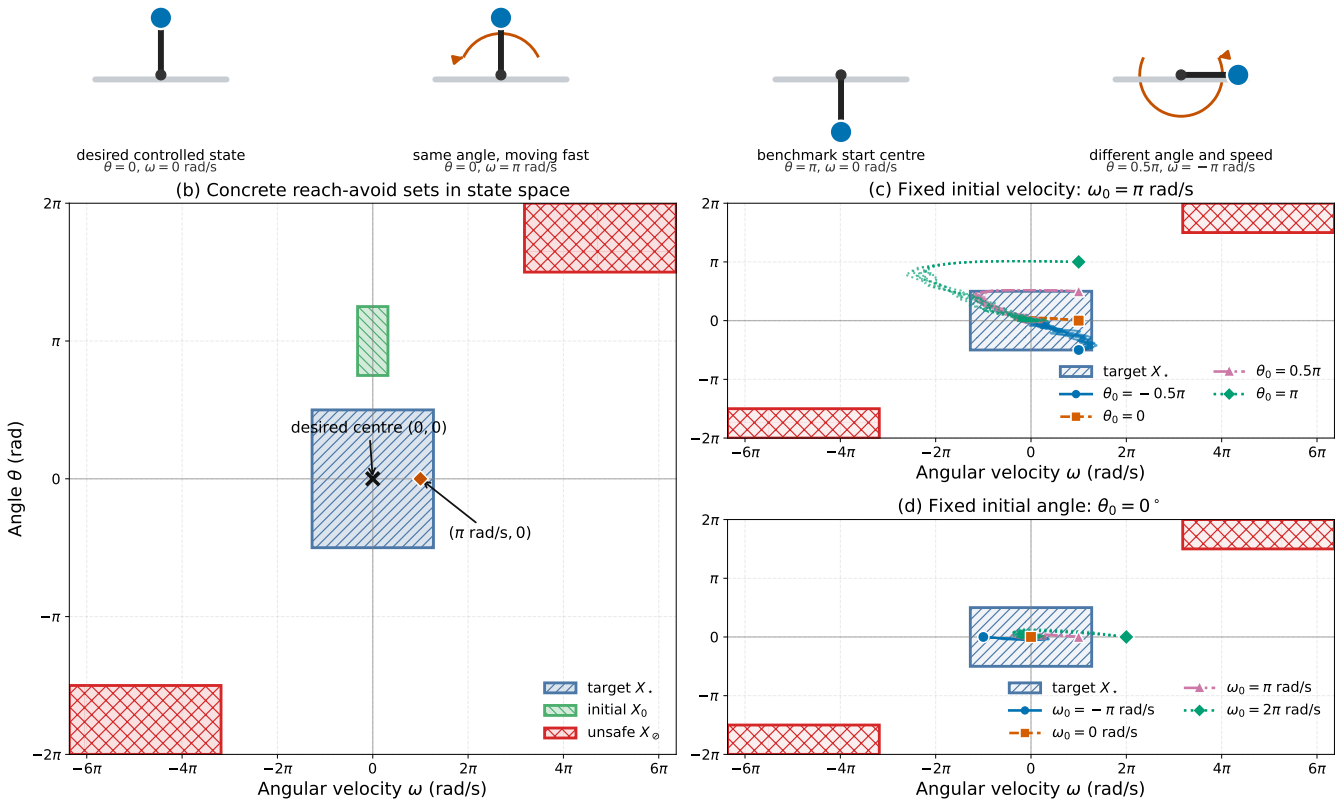

Figure 1: Running example used throughout the thesis: the stochastic inverted-pendulum benchmark. The upper row illustrates representative pendulum states. Panel (b) shows the concrete reach–avoid sets in state space: the initial set X_0, the target set $X_\star$, and the unsafe set $X_\varnothing$. Panels (c) and (d) show example trajectories in the (ω, θ)-plane for different initial conditions. The reach–avoid objective is to reach $X_\star$ before entering $X_\varnothing$.

If V is differentiable, its rate of change along trajectories of (4) is given by

$$\frac{d}{dt}V(x(t)) = \nabla V(x(t))^\top f(x(t)). \tag{5}$$

The right-hand side is the Lie derivative of V along the vector field f. It measures whether the dynamics move the state toward higher or lower values of V. If

$$\nabla V(x)^\top f(x) < 0 \qquad \text{for all } x \neq x^\star,$$

then the certificate value decreases strictly whenever the system is away from equilibrium. This provides a certificate of stability without requiring us to solve the differential equation explicitly (Khalil, 2002; Slotine and Li, 1991).

The physical intuition is that V behaves like a generalised energy function. In a mechanical system, friction or damping dissipates energy over time. Lyapunov theory abstracts this idea: instead of tracking physical energy directly, it constructs a mathematical energy-like function whose decrease proves stable behaviour. The idea of proving system properties by decreasing a scalar function makes Lyapunov functions the conceptual starting point for many subsequent certificate constructions.

Barrier certificates. The avoidance part of a reach–avoid objective can be grasped through barrier certificates. While Lyapunov functions are used to prove stability or convergence, barrier certificates are used to prove that trajectories starting from admissible initial states cannot enter an unsafe region (Prajna and Jadbabaie, 2004; Prajna et al., 2004; Ames et al., 2019). For a deterministic system

$$\dot{x} = f(x),$$

a standard zero-level-set formulation uses a scalar function $B : X \to \mathbb{R}$ whose zero level set

$$Z(B) := \{x \in X \mid B(x) = 0\}$$

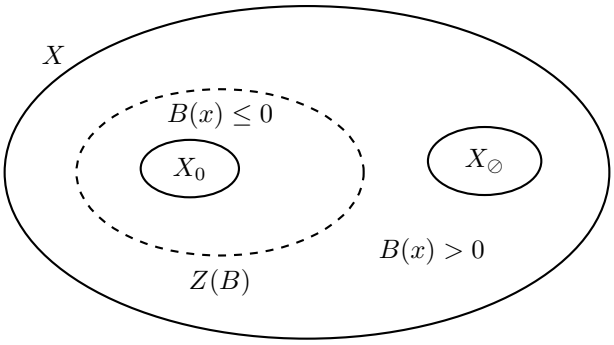

Figure 2: Schematic intuition behind a barrier certificate, following the standard zero-level-set view. The dashed curve $Z(B) = \{x \in X \mid B(x) = 0\}$ separates the initial set X_0, where $B(x) \leq 0$, from the unsafe set $X_\oslash$, where $B(x) > 0$.

separates admissible initial states from unsafe states. With the sign convention used here, a typical set of sufficient conditions is

$$B(x) \leq 0 \qquad \forall x \in X_0, \tag{6}$$

$$B(x) > 0 \qquad \forall x \in X_\oslash, \tag{7}$$

$$\nabla B(x)^\top f(x) < 0 \qquad \forall x \in Z(B). \tag{8}$$

The first two conditions place the initial set and unsafe set on different sides of the barrier. The last condition states that, on the separating boundary $Z(B)$, the certificate decreases along system trajectories. Hence, trajectories starting from the admissible initial set cannot cross the barrier into the unsafe region (Prajna and Jadbabaie, 2004; Prajna et al., 2004).

This intuition is illustrated in Figure 2. Barrier certificates thus share the same proof strategy as Lyapunov functions: both reason through a scalar function whose evolution along the dynamics implies a global guarantee. The difference is the guarantee obtained. Lyapunov functions certify stability or convergence toward an equilibrium, whereas barrier certificates certify safety by proving that trajectories cannot cross into the unsafe region.

A reach–avoid objective adds a second requirement. The system should not only avoid the unsafe set $X_\oslash$, but also reach a designated target set $X_\star$. Reach–avoid certification therefore combines two forms of reasoning: a safety argument separating the system from unsafe states, and a progress argument showing movement toward the target. In this sense, reach–avoid certificates combine barrier-style separation with Lyapunov-style decrease.

Running example. ***Stochastic inverted pendulum, continued.*** *In the pendulum example, a Lyapunov-style viewpoint would focus on convergence toward a desired controlled state. A barrier-style viewpoint instead focuses on excluding failure states, such as dangerous angle–velocity combinations. Reach–avoid certification combines both intuitions: the system should reach the target region, but it must do so before entering the unsafe region.*

Reach–avoid certificates. Reach–avoid certificates extend the previous ideas to specifications with both a target set and an unsafe set. The objective is not only to avoid $X_\oslash$, but to reach $X_\star$ before entering $X_\oslash$. A certificate for such an objective must therefore encode both separation and progress. It separates low-certificate initial states from high-certificate unsafe states, and it enforces decrease toward favourable regions until the target is reached.

This viewpoint is important for the rest of the thesis. The certificate value should not be interpreted as the true reach–avoid probability. Rather, it is an auxiliary scalar quantity from which a lower bound on the reach–avoid probability can be derived. The level structure of the certificate determines the certified probability guarantee.

From deterministic decrease to stochastic decrease in expectation. In stochastic systems, pointwise decrease is no longer the right notion. Random disturbances can temporarily increase the certificate even if the system is safe overall. What matters instead is how the certificate evolves *in expectation*. This leads to martingale and supermartingale reasoning (Blumenthal and Getoor, 1968; Prajna et al., 2007).

For a stochastic process $(V(X_t))_{t\geq 0}$, the supermartingale intuition is that the expected future value does not exceed the present value. In verification, one typically strengthens this to a strict expected decrease away from target states or safe subregions. This is the stochastic counterpart of Lyapunov decrease and is the main bridge between classical control theory and probabilistic verification.

2.3 Probabilistic reach-avoid certificates in discrete time

To understand the certificates used in this thesis, it is helpful to first consider the discrete-time stochastic reach-avoid setting. In that setting, Zikelic et al. (2023) introduced *reach-avoid supermartingales* (RASMs) as neural certificate functions for proving

formal reach-avoid guarantees for discrete-time nonlinear stochastic systems. This builds on the broader, older use of martingale and supermartingale arguments in probabilistic verification.

A discrete-time stochastic system can be written as

$$x_{t+1} = f(x_t, u_t, w_t), \qquad u_t = \pi(x_t), \tag{9}$$

where $w_t \sim \mu$ is a stochastic disturbance.

Definition 2 (Probabilistic reach–avoid specification)**.** *In thesis notation, a probabilistic reach–avoid specification is a tuple*

$$(X_\star, X_\oslash, X_0, \rho),$$

where $X_\star \subseteq X$ is the target set, $X_\oslash \subseteq X$ is the unsafe set, $X_0 \subseteq X$ is the initial set, and $\rho \in [0,1]$ is a probability threshold. The specification is satisfied if, for every $x_0 \in X_0$, the probability under the fixed policy and system dynamics of reaching $X_\star$ before entering $X_\oslash$ is at least ρ.

In that setting, a reach-avoid supermartingale is a certificate that is low on the initial set, high on the unsafe set, and decreases in expectation whenever the state is outside the target and below the unsafe certificate threshold.

Definition 3 (Discrete-time reach-avoid supermartingale, adapted from Zikelic et al. (2023))**.** *In simplified thesis notation, a discrete-time reach–avoid supermartingale is a continuous function $V : X \to \mathbb{R}$ such that:*

1. *$V(x) \geq 0$ for all $x \in X$;*
2. *$V(x) \leq 1$ for all $x \in X_0$;*
3. *$V(x) \geq \dfrac{1}{1-\rho}$ for all $x \in X_\oslash$, with $\rho \in [0,1)$;*
4. *there exists $\varepsilon > 0$ such that for all $x \in X \setminus X_\star$, either*

$$\mathbb{E}_{w\sim\mu}\big[V(f(x,\pi(x),w))\big] \leq V(x) - \varepsilon,$$

or

$$V(x) \geq \frac{1}{1-\rho}.$$

In particular, the threshold structure is

$$V(x) \leq 1 \quad \forall x \in X_0, \qquad V(x) \geq \frac{1}{1-\rho} \quad \forall x \in X_\oslash. \tag{10}$$

These two inequalities should be read as a level separation: the initial set is placed below the normalised level 1, whereas the unsafe set is placed above the higher level $1/(1-\rho)$. This is the discrete-time analogue of the general certificate levels α and β used later in the continuous-time formulation.

The certificate does two jobs at once. First, it establishes a *level separation*: initial states are placed in a low-value region and unsafe states in a high-value region. Second, it establishes a *direction of progress*: until the target is reached, the expected certificate value decreases. Together, these properties imply that the system is likely to move toward the target before accumulating enough increase to reach the unsafe level.

Running example. ***Stochastic inverted pendulum, continued.*** *If the pendulum were analysed in discrete time, one would reason about the expected certificate value after one transition step. This is the setting of discrete-time reach-avoid supermartingales. The present thesis instead considers the continuous-time version, where the system evolves at every instant and the one-step decrease condition is replaced by an infinitesimal one.*

The main takeaway is the level-set structure. Initial states are placed below a low certificate level, unsafe states are placed above a high certificate level, and outside the target the certificate is required to decrease in expectation. This combination allows a probability guarantee to be derived without estimating the exact reach–avoid probability directly. Continuous-time certificates use the same level-set intuition, but replace the one-step expected decrease condition by an infinitesimal generator condition.

2.4 Continuous-time stochastic systems and the infinitesimal generator

The main technical shift from discrete time to continuous time is that the expected one-step decrease is replaced by a local infinitesimal condition. For a continuous-time stochastic system of the form (1), the relevant object is the *infinitesimal generator*. For sufficiently smooth V, the generator of the closed-loop dynamics is

$$(G_\pi V)(x) = f_\pi(x)^\top \nabla V(x) + \frac{1}{2}\operatorname{Tr}\big(g_\pi(x) g_\pi(x)^\top \nabla^2 V(x)\big). \tag{11}$$

The first term is the deterministic contribution coming from the drift. The second term is the stochastic correction coming from the diffusion and the curvature of V. In deterministic continuous-time systems, only the first-order term appears. In stochastic systems, the second-order term is essential. This is one of the reasons why deterministic or naively discretised certificate techniques do not directly transfer to continuous-time stochastic dynamics (Neustroev et al., 2025).

Intuitively, $(G_\pi V)(x)$ measures the instantaneous expected rate of change of the certificate at state x. Thus, the continuous-time analogue of "the certificate decreases in expectation" is a generator inequality of the form

$$(G_\pi V)(x) \leq -\zeta, \tag{12}$$

for some $\zeta > 0$, on the part of the state space where a decrease is required.

Running example. ***Stochastic inverted pendulum, continued.*** *For the pendulum, the drift f_π describes the nominal angle–velocity evolution under the controller, while the diffusion g_π describes the stochastic disturbance. The generator combines both effects. The drift term measures how the controller moves the state across the certificate landscape, while the diffusion term captures how noise interacts with the curvature of that landscape. This is why continuous-time stochastic certificates need the second-order term in* (11)*.*

This infinitesimal viewpoint is the basis of the framework of Neustroev et al. (2025), which introduces neural supermartingale certificates for continuous-time stochastic systems.

Definition 4 (Probabilistic reach–stay–avoid specification)**.** *In the full framework of Neustroev et al. (2025), a probabilistic reach–stay–avoid specification is a tuple*

$$(X_\star, X_\oslash, X_0, \varepsilon, \delta),$$

where, in this definition, $X_\star$ and $X_\oslash$ denote the possibly time-dependent families $(X_\star(t))_{t\geq 0}$ and $(X_\oslash(t))_{t\geq 0}$, respectively, X_0 is the initial set, $\varepsilon \in [0, 1)$ is a reach–avoid probability, and $\delta \in [0, 1)$ is a stay probability. Intuitively, the requirement is to reach the target while avoiding the unsafe set with probability at least ε, and then remain in the target with probability at least δ.

The corresponding certificate theorem of Neustroev et al. (2025, Theorem 1) uses four certificate levels, two for reach–avoid and two for stay, and derives the probabilities through ratio formulas of the form

$$\varepsilon = 1 - \frac{\alpha_{\mathrm{RA}}}{\beta_{\mathrm{RA}}}, \qquad \delta = 1 - \frac{\alpha_{\mathrm{S}}}{\beta_{\mathrm{S}}}. \tag{13}$$

Moreover, Neustroev et al. (2025, Corollary 1) show that the reach–avoid part can be certified on its own by omitting the stay conditions. In the present thesis, we use precisely this time-homogeneous reach–avoid fragment of their framework, which is formalised in Chapter 3.

2.5 Continuous-time reach-avoid certificates

The certificate framework of Neustroev et al. (2025) provides the direct continuous-time parent of this thesis. In the simplified, time-homogeneous reach–avoid setting adopted later in Chapter 3, a certificate $V : X \to \mathbb{R}_{\geq 0}$ is associated with parameters $0 < \alpha < \beta$ and $\zeta > 0$ such that

$$V(x) \leq \alpha \qquad \forall x \in X_0, \tag{14}$$

$$V(x) \geq \beta \qquad \forall x \in X_\oslash, \tag{15}$$

$$(G_\pi V)(x) \leq -\zeta \qquad \forall x \in X \setminus X_\star \text{ such that } V(x) < \beta. \tag{16}$$

The meaning of these conditions corresponds to the discrete-time case, but with the generator replacing the one-step expectation. The initial set lies below the lower threshold α, the unsafe set lies above the upper threshold β, and all states below the unsafe threshold but outside the target satisfy a strict expected decrease condition. Those inequalities imply a lower bound on the reach-avoid probability. In the notation used later in this thesis, the certified bound is

$$\rho = 1 - \frac{\alpha}{\beta}. \tag{17}$$

This formula is central. It shows that the guarantee is determined not by the absolute magnitude of the certificate, but by the relative separation between the initial-set and unsafe-set levels. If the unsafe threshold β is much larger than the initial threshold α, then the certified reach-avoid probability is high. There is also an important conceptual point here. The certificate V is not, in general, a probability function. Rather, it is an auxiliary scalar witness from which a probability bound can be derived. This distinction is useful later when discussing reclaiming: VeRecycle does not estimate the exact reach–avoid probability of the modified system. Instead, it computes the reach–avoid lower bound that can still be certified by reusing the same fixed certificate.

Running example. ***Stochastic inverted pendulum, continued.*** *For the pendulum, the inequalities in* (14)–(16) *say the following. The allowed initial states must lie below the lower certificate level α. The unsafe angle–velocity states must lie above the higher level β. Everywhere outside the target where the certificate is still below β, the generator must be negative. Thus, before the system reaches the target, the certificate is forced to decrease in expectation.*

2.6 Neural certificates

For nonlinear systems and neural-network controllers, certificates are rarely available in closed form. This motivates *neural certificates*: certificates represented using neural networks and trained by combining learning with sound verification (Chang et al., 2019; Abate et al., 2021a; Dawson et al., 2023; Neustroev et al., 2025).

The general workflow is as follows. A neural network V_θ is trained to satisfy differentiable relaxations of the certificate conditions on sampled states. A verification procedure then checks whether the trained candidate certificate satisfies the required inequalities globally over the relevant subsets of the state space. If verification fails, the violating regions are fed back as counterexamples, and the certificate is retrained (Chatterjee et al., 2023; Neustroev et al., 2025). This learner-verifier loop continues until either a valid certificate is found or the search terminates.

The learning step is expressed through a loss function that penalises violations of the certificate conditions on sampled states. In the full reach–avoid–stay framework of Neustroev et al. (2025), the training loss is written as

$$L := L_\oslash + L_0 + L_\star + L_\downarrow + R,$$

where $L_\oslash$ penalises violations of the unsafe-set condition, L_0 penalises violations of the initial-set condition, $L_\star$ enforces the goal condition used for the stay part of the specification, $L_\downarrow$ penalises violations of the generator decrease condition, and R is a regularisation term.

The present thesis uses only the reach–avoid fragment of this framework. By Neustroev et al. (2025, Corollary 1), the reach–avoid property can be certified without the stay property by omitting the stay-specific conditions. Therefore, the stay levels α_S, β_S and the corresponding goal/stay loss terms are not part of the formal certificate used here.

Aligned with Chapter 3, the reach–avoid certificate uses levels $0 < \alpha < \beta$, generator margin $\zeta > 0$, initial set X_0, unsafe set $X_\oslash$, and target set $X_\star$. Schematically, the reach–avoid training loss can be written as

$$L_{\mathrm{RA}} := L_0 + L_\oslash + L_\downarrow + R,$$

where L_0 penalises violations of $V(x) \le \alpha$ on X_0, $L_\oslash$ penalises violations of $V(x) \ge \beta$ on $X_\oslash$, $L_\downarrow$ penalises violations of $(G_\pi V)(x) \le -\zeta$ on $L_\beta^-(V) \setminus X_\star$, and R is a regularisation term. The target set still appears in $L_\downarrow$ because the reach–avoid decrease condition is only required before the target is reached. The nonnegativity condition is not written as a separate loss term because the implementation enforces it architecturally through a nonnegative network output.

Interval-bound propagation (IBP) is one way to perform the sound bounding step used in such verification procedures. Instead of evaluating the neural certificate only at sampled points, the verifier covers the relevant subset of the state space by axis-aligned boxes. For each box, IBP propagates input intervals through the neural network to obtain lower and upper bounds on all possible certificate values inside that box. In the continuous-time setting, the same idea is also used to bound the generator term $G_\pi V$. If the resulting bounds imply the required certificate inequalities on every box, then the condition is verified on the whole covered region; otherwise, the box can be refined or returned as a counterexample (Neustroev et al., 2025).

Running example. ***Stochastic inverted pendulum, continued.*** *In the pendulum benchmark, the certificate is represented by a neural network over the two-dimensional state space. During training, the network is encouraged to be low near the desired target region, high near unsafe regions, and to satisfy the generator decrease condition where required. Verification then checks whether these properties hold over the full relevant state space, not only at the sampled training points.*

Figure 3 illustrates this learner-verifier intuition on the stochastic inverted-pendulum benchmark. Each panel should be read as a certificate landscape over the state space $x = (\omega, \theta)$. The colours indicate the learned certificate value, while the rectangles mark the relevant reach–avoid regions, such as the initial set X_0, the target set $X_\star$, and the unsafe set $X_\oslash$. Across training rounds, the landscape becomes more structured: unsafe regions are pushed toward higher certificate values, while regions from which the target can be reached safely become more clearly separated. The figure is not a proof of correctness; it only illustrates how the learner shapes the candidate certificate before the verifier checks the inequalities soundly.

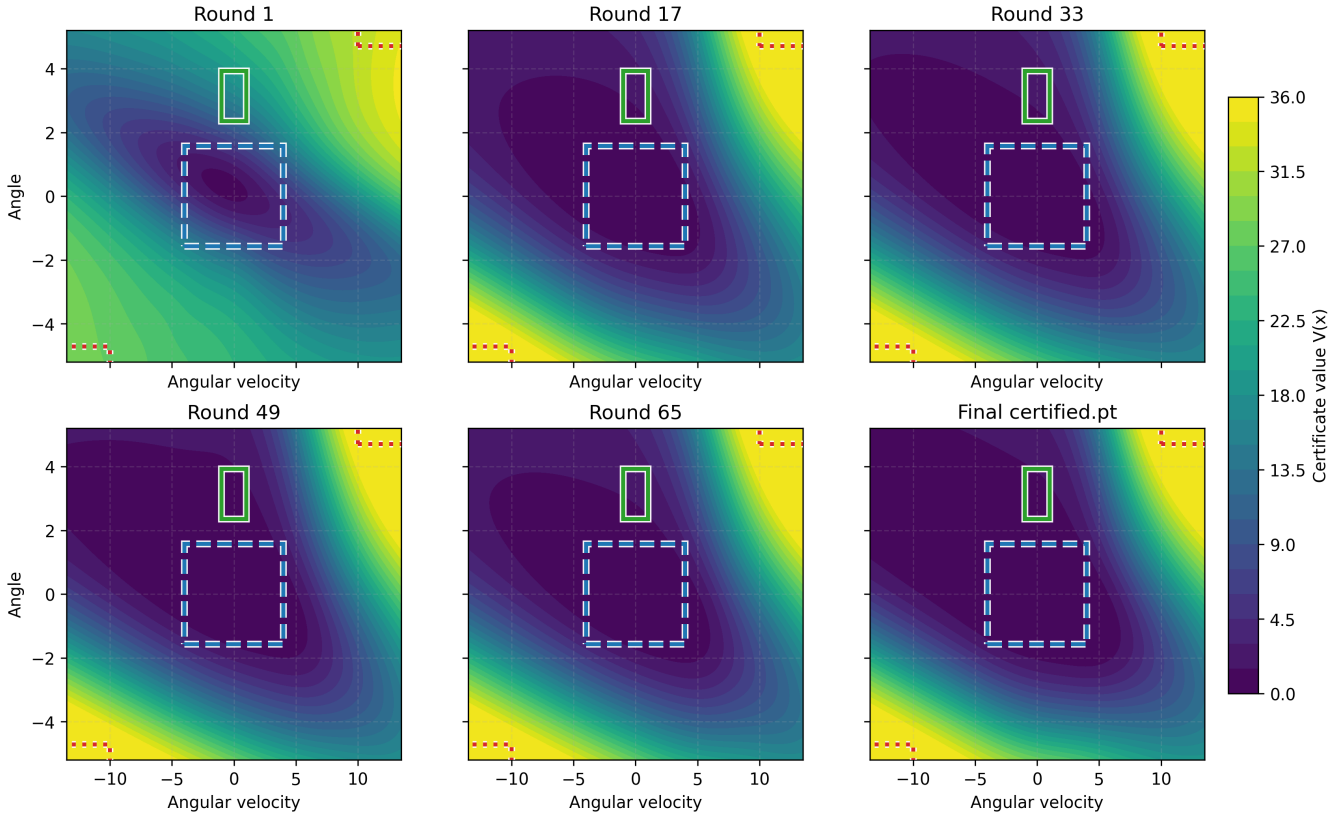

Figure 3: Neural certificate landscape over training snapshots for the stochastic inverted-pendulum benchmark. Each panel shows V_θ over the state space $x = (\omega, \theta)$. The rectangles indicate the reach–avoid sets X_0, $X_\star$, and $X_\oslash$. Across training, the certificate landscape becomes more structured, separating favourable regions from unsafe ones. Continuous-time VeRecycle starts only after such a certificate has been certified and fixed.

In continuous-time stochastic settings, the same learner–verifier idea applies, but the certificate inequalities are expressed through the infinitesimal generator in (11). Thus, after training a candidate certificate, verification checks sound bounds on both V and $G_\pi V$ over the relevant subsets of the state space, using methods such as interval-bound propagation or related symbolic techniques (Neustroev et al., 2025). In this thesis, this point is important because continuous-time VeRecycle starts from an already-certified certificate V: the certificate is kept fixed, and the reclaiming step asks how much of its original guarantee can still be certified after a localised model change.

2.7 Localised model change and the motivation for reclaiming

Once a certificate has been computed for a fixed system and policy, the guarantee it provides is tied to the dynamics for which the certificate conditions were verified. In practice, however, the deployed system may experience changes that are not captured by the original model. Examples include altered friction, changed disturbances, unanticipated obstacles, or other local effects. If such changes affect only a known subset of the state space, it is natural to ask whether the original certificate can still be partially reused instead of recomputing a new one from scratch.

Following the localised-change setting of Lutz et al. (2025), let $X_q \subseteq X$ denote the region in which the dynamics may have changed. A localised model change means that the modified dynamics agree with the original dynamics on $X \setminus X_q$, while they may differ inside X_q. Outside X_q, the modified dynamics are assumed to agree with the original dynamics. Inside X_q, the modified dynamics may differ from the original model within the admissible class of local changes considered later in Chapter 3.

The key idea of VeRecycle is that the amount of guarantee recoverable from a fixed certificate is governed by the certificate infimum over the changed region. In the notation used throughout this thesis, this quantity is

$$m := \inf_{x \in X_q} V(x).$$

This corresponds to the infimum of the original certificate over the changed region in the notation of Lutz et al. (2025). Intuitively, m measures the lowest certificate level attained in the part of the state space where the dynamics may have changed. If V is large throughout X_q, then the changed region already lies above a high certificate level, so the proof can avoid relying on the

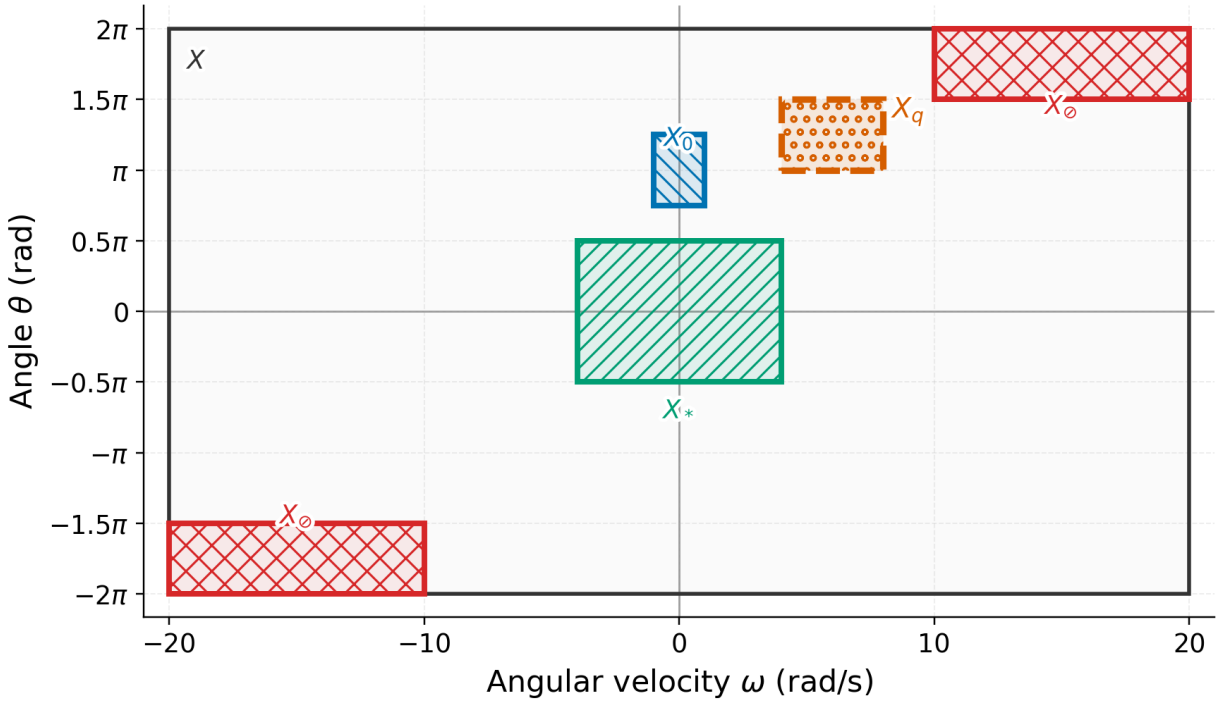

Figure 4: Schematic illustration of a localised model change in the stochastic inverted-pendulum running example. The plot shows the pendulum state space in terms of angular velocity ω and angle θ, together with the domain X, the admissible initial set X_0, the target set $X_\star$, and the unsafe regions $X_\oslash$. The orange region X_q marks the part of the state space where the model may change, for example because the actuator produces less torque than the nominal model predicts. Outside X_q, the original dynamics are assumed to remain valid; inside X_q, the drift or diffusion may differ from the nominal model.

original decrease condition there while still preserving part of the guarantee. If V is small somewhere in X_q, then the changed region overlaps with a low-level part of the certificate landscape, and the guarantee recoverable from the fixed certificate may degrade substantially.

Running example. ***Stochastic inverted pendulum, continued.*** *In the pendulum example, a localised change corresponds to a restricted part of the (ω, θ)-plane where the model may no longer be reliable. For instance, the actuator may produce less torque than the nominal model predicts only in the orange region X_q shown in Figure 4. This region is denoted by X_q. Outside X_q, the original pendulum dynamics are assumed to remain valid; inside X_q, the drift or diffusion may differ from the nominal model within the admissible class of local changes. Thus, the value $m = \inf_{x \in X_q} V(x)$ measures the certificate infimum over precisely the part of the state space where the dynamics may have changed. Figure 4 illustrates this idea schematically.*

The next subsection recalls the discrete-time VeRecycle result more precisely. For this thesis, the key insight is that the recovered guarantee is controlled by a local certificate infimum on the changed region.

2.8 The VeRecycle framework in discrete time

The previous subsection explained *why* reclaiming is needed. We now briefly describe *how* VeRecycle addresses that problem in the discrete-time setting.

VeRecycle is introduced by Lutz et al. (2025) as a framework for reclaiming probabilistic reach-avoid guarantees after a localised change in the system dynamics. Its input is an already-certified policy together with its original probabilistic reach-avoid certificate, and it assumes that the new dynamics may differ from the original ones only inside a known subset $X_q \subseteq X$, translated here into the notation of this thesis. Importantly, VeRecycle does *not* require a full model of the new dynamics inside X_q. Instead, it asks for the largest threshold that can be certified by reusing the same certificate, uniformly over all modified dynamics that agree with the original dynamics outside X_q. This makes the framework robust to unknown local changes and avoids immediate re-certification of the full system (Lutz et al., 2025).

Problem solved by VeRecycle. In the notation used throughout this thesis, one starts from an original certified threshold ρ and a certificate V. A threshold $\rho' \in [0, \rho]$ is called reclaimable if the same certificate V still certifies the reach–avoid specification with threshold ρ' for every admissible local modification whose deviation from the original dynamics is confined to X_q. The goal is then to determine the maximum reclaimable threshold, denoted $\tilde{\rho}$.

Exact reclaiming formula. The main theoretical result of VeRecycle is that the exact reclaimable threshold depends only on two quantities: the original certified threshold ρ, and the infimum of the original certificate over the changed region,

$$m := \inf_{x \in X_q} V(x).$$

Translated into the notation of this thesis, if $m \geq 1$, then the reclaimable threshold is

$$\tilde{\rho} = \min\left(\rho,\, 1 - \frac{1}{m}\right). \tag{18}$$

If $m < 1$, then no reclaimable threshold exists under the discrete-time VeRecycle definition. In other words, VeRecycle shows that the reclaiming problem collapses to a local certificate quantity: the infimum of V over the changed region X_q (Lutz et al., 2025, Problem 1, Theorem 1).

Here, the threshold 1 appears because the discrete-time reach–avoid certificate of Lutz et al. (2025) is normalised so that the initial set lies below certificate level 1. In the continuous-time formulation of this thesis, the corresponding lower certificate level is denoted by α.

Why this formula is intuitive. This formula has a clear interpretation. If the certificate is large on X_q, then the changed region already lies above a high certificate level. In that case, the proof does not need to rely on the original decrease condition inside X_q, and a nontrivial reach–avoid lower bound can still be certified. If the certificate takes small values somewhere in X_q, then the changed region overlaps with a low-level part of the certificate landscape. Since the dynamics may be arbitrary there, reusing the fixed certificate can then only certify a weaker lower bound, or no nontrivial bound at all.

The proof idea in Lutz et al. (2025) captures this intuition exactly by proving both directions of the equality in eq. (18). First, the formula is shown to be attainable: the reclaimed threshold is chosen so that all states in X_q already lie above the relevant certificate level, meaning that the expected decrease no longer needs to be relied on there. Second, the formula is shown to be maximal: by considering fully absorbing dynamics inside X_q, Lutz et al. (2025) show that no larger threshold can be guaranteed uniformly for all admissible local changes (Lutz et al., 2025, Lemmas 1–3).

Running example. ***Stochastic inverted pendulum, continued.*** *Translated to the pendulum setting, the VeRecycle question is the following: suppose the original certificate V was valid for the nominal pendulum dynamics, but the dynamics may now be different inside a known angle–velocity region X_q. Without relying on the modified dynamics inside X_q, can the same certificate still prove a reach–avoid guarantee? VeRecycle says that the answer depends on how low V becomes inside X_q, namely on $m = \inf_{x \in X_q} V(x)$.*

Sound approximation in practice. The exact infimum of a neural certificate is generally expensive to compute. VeRecycle therefore also uses a sound approximation procedure based on lower and upper bounds on the infimum. Concretely, the changed region X_q can be covered by boxes, and IBP can be used to compute certified lower and upper bounds on V over each box. If

$$1 \leq I^- \leq \inf_{x \in X_q} V(x) \leq I^+,$$

then one obtains certified lower and upper bounds on the reclaimable threshold:

$$\tilde{\rho}^- = \min\left(\rho,\, 1 - \frac{1}{I^-}\right), \qquad \tilde{\rho}^+ = \min\left(\rho,\, 1 - \frac{1}{I^+}\right). \tag{19}$$

VeRecycle outputs $\tilde{\rho}^-$ as a guaranteed reclaimed threshold. In the implementation of Lutz et al. (2025), such bounds are obtained using interval bound propagation and mesh refinement, which makes reclaiming much cheaper than retraining and re-verifying a new certificate from scratch (Lutz et al., 2025, Theorem 2).

Why VeRecycle matters for this thesis. For the present thesis, the key contribution of VeRecycle is not only the discrete-time formula in eq. (18), but the structural insight behind it: after a localised change, the maximum guarantee reusable from the fixed certificate is determined by the interaction between the original certified threshold and a local certificate infimum on the changed region. This is precisely the insight that motivates the continuous-time development later in this thesis.

2.9 From discrete-time VeRecycle to the present thesis

The present thesis is motivated by the structural similarity between the discrete-time VeRecycle idea and the continuous-time reach–avoid certificate framework. In both settings, a fixed certificate establishes a reach–avoid guarantee via two ingredients: boundary separation between the initial and unsafe states, and a decrease condition away from the target. In both settings, a localised model change can invalidate the decrease condition only inside the changed region, while leaving the certificate unchanged elsewhere.

This suggests the central question of the thesis: can the infimum-based reclaiming principle of VeRecycle be extended from discrete-time stochastic systems to continuous-time stochastic systems certified through the generator-based proof rule of Neustroev et al. (2025)?

Chapter 3 formulates that question precisely in the time-homogeneous reach–avoid setting used in this thesis. There, the original certified reach–avoid bound is written as

$$\rho = 1 - \frac{\alpha}{\beta}, \tag{20}$$

and the changed-region certificate infimum is

$$m := \inf_{x \in X_q} V(x). \tag{21}$$

The next chapters show that, in continuous time, the maximal reclaimed bound induced by a fixed certificate V is again governed by this local quantity m. More precisely, Chapter 4 proves that the reclaimed bound is characterised by the interaction between the original bound ρ and the changed-region infimum m, yielding the continuous-time analogue of the VeRecycle rule.

2.10 Chapter summary

This chapter introduced the background needed for the remainder of the thesis. First, it reviewed certificate-based verification as an alternative to trajectory-wise reasoning in stochastic dynamical systems. Second, it explained the progression from Lyapunov functions to barrier certificates and then to stochastic supermartingale certificates. Third, it discussed discrete-time probabilistic reach–avoid certificates, which provide the main intuition for how level-set reasoning yields a probability bound. Fourth, it introduced the continuous-time stochastic setting and the infinitesimal generator, which is the key technical tool behind the certificate framework used in this thesis. Finally, it presented the motivation for reclaiming guarantees under localised model change and the discrete-time VeRecycle idea that inspires the present work. The next chapter builds on this background by fixing the system model, notation, and reach–avoid objective used throughout the thesis, and by formalising the reclaiming problem for continuous-time stochastic certificates.

3 Problem Formulation

This chapter presents the system model, the reach–avoid goal, and the reasoning structure used to study reclaimed guarantees under localised model changes.

Reader guide. This chapter fixes the five ingredients used throughout the remainder of the thesis: the closed-loop stochastic dynamics, the reach-avoid objective, the certificate conditions, the model of localised change, and the notion of reclaimed guarantees. To make the definitions concrete, the stochastic inverted pendulum introduced in Chapter 2 is continued as a running example. Readers mainly interested in the main theorem of Chapter 4 may focus on eqs. (25) and (27) to (31).

Compared with the background chapter, the purpose here is not to present new intuition, but to fix the exact continuous-time objects that are extended by the reclaiming result in Chapter 4: the generator-based certificate, the admissible localised change model, and the notion of a universally valid reclaimed guarantee.

3.1 Notation

Symbol	Description
$X \subset \mathbb{R}^\ell$	Compact state space
$x \in X$	Individual state
$x_0 \in X_0$	Initial state
X_t	State of the stochastic process at time t
W_t	Wiener process driving the stochastic disturbance
$X_0 \subset X$	Initial set
$X_\star \subset X$	Target set
$X_\oslash \subset X$	Unsafe set
$X_q \subset X$	Region of localised model change
U	Control space
u_t	Control input applied at time t
$\pi : X \to U$	Fixed feedback policy mapping states to controls
$f(x,u)$, $g(x,u)$	Open-loop drift and diffusion functions
f_π, g_π	Drift and diffusion functions of the original closed-loop SDE
$\tilde{f}_\pi$, $\tilde{g}_\pi$	Drift and diffusion functions under modified dynamics
G_π	Infinitesimal generator under the original dynamics
$\tilde{G}_\pi$	Infinitesimal generator under the modified dynamics
$V : X \to \mathbb{R}_{\geq 0}$	Continuous-time reach-avoid certificate
α	Upper certificate level allowed on the initial set X_0
β	Lower certificate level required on the unsafe set $X_\oslash$, with $0 < \alpha < \beta$
$\zeta > 0$	Generator decrease margin
$m := \inf_{x \in X_q} V(x)$	Infimum of the certificate over the change region
τ	First hitting time of $X_\star \cup X_\oslash$ under the dynamics currently under consideration
$\mathbb{P}_{x_0}$	Probability measure induced by the original SDE starting at x_0
$\mathbb{P}_{x_0}^{(\tilde{f}_\pi, \tilde{g}_\pi)}$	Probability measure induced by the modified SDE starting at x_0
ρ_{rec}	Reclaimed reach-avoid guarantee certifiable by reusing V
$\rho^\star_{\mathrm{rec}}(V)$	Maximal reclaimed guarantee certifiable by reusing V
ρ	Original certified reach-avoid bound, defined by $\rho := 1 - \alpha/\beta$
$\rho_{\mathrm{rec}}(V)$	Candidate reclaimed reach-avoid bound induced by V and m

How to read the notation. Throughout the thesis, sets of states are denoted by uppercase symbols such as X, X_0, $X_\star$, $X_\oslash$, and X_q. Individual states are denoted by lowercase symbols such as x and x_0. The symbol X_t is the exception to this set notation: it denotes the random state of the stochastic process at time t, whereas x_0 denotes a fixed initial state. Symbols with a tilde refer to modified dynamics after a localised change. The quantity m will play a central role later: it is the smallest certificate value on the changed region and determines how much of the original guarantee can still be reclaimed.

3.2 System model

Conceptual overview. We study the safety of dynamical systems in which the state evolves continuously over time under uncertainty. A dynamical system specifies how the current state determines future behaviour, generally through differential equations.

In this setting, we are interested in *guarantees*: statements that specify what the system will do under all possible behaviours. For example, a reach-avoid guarantee ensures that, starting from a given set of initial states, the system reaches a target region before entering an unsafe region with at least a specified probability (Zikelic et al., 2023; Neustroev et al., 2025).

Obtaining such guarantees is challenging. Simulations can only explore a finite number of trajectories, while the system may exhibit infinitely many possible behaviours due to stochasticity. As a result, simulation-based validation cannot provide formal assurance (Baier and Katoen, 2008; Clarke et al., 2018).

To address this, we use *certificate functions*. A certificate assigns a scalar value to each state, which can be interpreted as a risk measure. Instead of analysing individual trajectories, certificates allow us to verify that the system progresses in a safe direction in expectation (Kalman and Bertram, 1960; Prajna et al., 2004; Neustroev et al., 2025).

From such a certificate, we can derive a probabilistic guarantee on the reach-avoid objective. In this way, certificates serve as a tractable tool for obtaining guarantees that hold uniformly over all possible system behaviours (Neustroev et al., 2025, Theorem 1 and Corollary 1).

Running example. ***Stochastic inverted pendulum.*** *Throughout this chapter, we continue the stochastic inverted pendulum example from Chapter 2. The pendulum can be viewed as a simple robotic control system: its state evolves continuously over time, it is controlled through a feedback policy, and it is affected by stochastic disturbances. A natural state description is*

$$x = (\omega, \theta),$$

where ω is angular velocity and θ is the pendulum angle. The control objective is to steer the pendulum toward a desired controlled region while avoiding failure states.

The example is used only to build intuition. The formal development below is general and does not depend on the pendulum specifically. However, the pendulum gives a concrete interpretation to the abstract objects used later: the stochastic differential equation models the physical evolution of the robotic system, the target set corresponds to successful control, the unsafe set corresponds to failure, and the certificate V acts as a risk map over the angle–velocity state space.

Control policy. We restrict attention to the closed-loop system induced by a fixed feedback policy

$$\pi : X \to U,$$

where U denotes the control space. Thus, when the system is in state X_t, the applied control input is

$$u_t = \pi(X_t).$$

Starting from the controlled stochastic dynamics

$$dX_t = f(X_t, u_t)\, dt + g(X_t, u_t)\, dW_t,$$

the induced closed-loop dynamics can therefore be written as

$$dX_t = f_\pi(X_t)\, dt + g_\pi(X_t)\, dW_t,$$

where

$$f_\pi(x) := f(x, \pi(x)), \qquad g_\pi(x) := g(x, \pi(x)).$$

Running example. ***Stochastic inverted pendulum, continued.*** *For the pendulum, the state is $x = (\omega, \theta)$, where ω is angular velocity and θ is the angle. The fixed feedback policy maps this state to the torque input*

$$u = \pi(\omega, \theta).$$

After substituting this policy into the pendulum dynamics, the closed-loop drift and diffusion are

$$f_\pi(\omega, \theta) = f((\omega, \theta), \pi(\omega, \theta)), \qquad g_\pi(\omega, \theta) = g((\omega, \theta), \pi(\omega, \theta)).$$

The verification problem in this thesis is therefore not to synthesise a new torque controller, but to reason about the reach–avoid guarantee of this already fixed closed-loop pendulum system.

Dynamics. We consider the stochastic differential equation

$$dX_t = f_\pi(X_t)\, dt + g_\pi(X_t)\, dW_t. \tag{22}$$

The functions f_π and g_π are assumed to be Lipschitz continuous and continuously differentiable, ensuring a unique strong solution for every initial state (Øksendal, 1998, Chapter 5).

For twice continuously differentiable functions $V : X \to \mathbb{R}_{\geq 0}$, the infinitesimal generator describes the expected instantaneous change of V along the system trajectories:

$$(G_\pi V)(x) = f_\pi(x)^\top \nabla V(x) + \tfrac{1}{2}\mathrm{Tr}\big(g_\pi(x) g_\pi(x)^\top \nabla^2 V(x)\big)\,. \tag{23}$$

Running example. ***Stochastic inverted pendulum, continued.*** *In the pendulum state space, the drift describes the nominal angle–velocity evolution under the fixed policy, while the diffusion describes the stochastic disturbance. The generator $G_\pi V$ measures the instantaneous expected change of the certificate landscape along these stochastic closed-loop dynamics.*

3.3 Reach-avoid objective

A reach-avoid problem is specified by an initial set X_0, a target set $X_\star$, and an unsafe set $X_\oslash$, where $X_\star$ and $X_\oslash$ are disjoint.

The objective is for the system to reach the target before entering the unsafe set. This is formalised via the stopping time

$$\tau := \inf\{t \geq 0 \mid X_t \in X_\star \cup X_\oslash\}.$$

The reach-avoid objective is satisfied if $X_\tau \in X_\star$. The corresponding probability is given by

$$\mathbb{P}_{x_0}(X_\tau \in X_\star).$$

Running example. ***Stochastic inverted pendulum, continued.*** *For the pendulum, the initial set X_0 contains the states from which we want the guarantee to hold. The target set $X_\star$ represents the desired controlled region, while the unsafe set $X_\oslash$ represents failure configurations, for example angle–velocity combinations from which safe recovery is not certified. The reach-avoid probability $\mathbb{P}_{x_0}(X_\tau \in X_\star)$ therefore measures how likely it is that the pendulum reaches the desired region before failing.*

3.4 Reach-avoid certificates

We now specialise the continuous-time certificate framework to the time-homogeneous reach–avoid setting used in this thesis. Following Neustroev et al. (2025), the certificate conditions below form the proof rule from which the reach–avoid guarantee is derived.

Definition 5 (Continuous-time reach-avoid certificate)**.** *A function $V : X \to \mathbb{R}_{\geq 0}$ is called a continuous-time reach-avoid certificate if there exist constants $0 < \alpha < \beta$ and $\zeta > 0$ such that:*

1. *$V(x) \leq \alpha$ for all $x \in X_0$,*
2. *$V(x) \geq \beta$ for all $x \in X_\oslash$,*
3. *$(G_\pi V)(x) \leq -\zeta$ for all $x \in X \setminus X_\star$ such that $V(x) < \beta$.*

This formulation captures the same proof idea as the continuous-time certificate framework of Neustroev et al. (2025), while restricting attention to the time-homogeneous reach-avoid case considered in this thesis. This is in line with the VeRecycle formulation in discrete time (Lutz et al., 2025).

Interpretation. The function V assigns a scalar value to each state that indicates its risk level according to the certificate. Low values correspond to states that the certificate treats as safer, while high values correspond to states closer to violating the reach-avoid objective. The certificate asserts that, as long as $V(x) < \beta$ and the target has not yet been reached, the expected instantaneous evolution of the system satisfies the required decrease condition.

Rather than checking individual trajectories, we verify that the system consistently moves toward lower certificate values in expectation. This avoids the need to analyse individual trajectories, which is generally intractable under stochastic dynamics, and instead provides guarantees that hold uniformly over all possible behaviours.

Running example. ***Stochastic inverted pendulum, continued.*** *For the pendulum, the certificate V can be interpreted as a risk landscape over the angle–velocity plane. The certificate conditions ensure that trajectories starting in the initial set begin in a low-value region, that unsafe states lie beyond a higher threshold, and that, before reaching the target, the expected evolution of the system moves toward lower certificate values. This is the intuition behind why a single scalar function can certify a probabilistic reach-avoid guarantee.*

Figure 5 makes this viewpoint concrete for the running pendulum example. The left panel shows representative closed-loop trajectories under the original dynamics. The right panel shows the certificate landscape over the same angle–velocity state space. Rather than reasoning directly about all stochastic trajectories, the certificate assigns a scalar value to each state. The shaded region X_q indicates where the dynamics may later differ from the original model.

Guarantee. The existence of a reach–avoid certificate implies the following probabilistic guarantee (Neustroev et al., 2025, Theorem 1 and Corollary 1):

$$\mathbb{P}_{x_0}(X_\tau \in X_\star) \geq 1 - \frac{\alpha}{\beta}, \qquad \forall x_0 \in X_0. \tag{24}$$

3.5 Localised model changes

The above guarantees assume that the system dynamics remain unchanged. In practice, local changes may occur, for example due to variations in friction, torque response, or stochastic disturbances. In the pendulum example, this means that the dynamics may be unreliable only for certain angle–velocity ranges.

Following the localised-change setting of Lutz et al. (2025), we model this by allowing the dynamics to change within a subset $X_q \subseteq X$, while remaining unchanged elsewhere:

$$\tilde{f}_\pi(x) = f_\pi(x), \qquad \tilde{g}_\pi(x) = g_\pi(x), \qquad x \in X \setminus X_q. \tag{25}$$

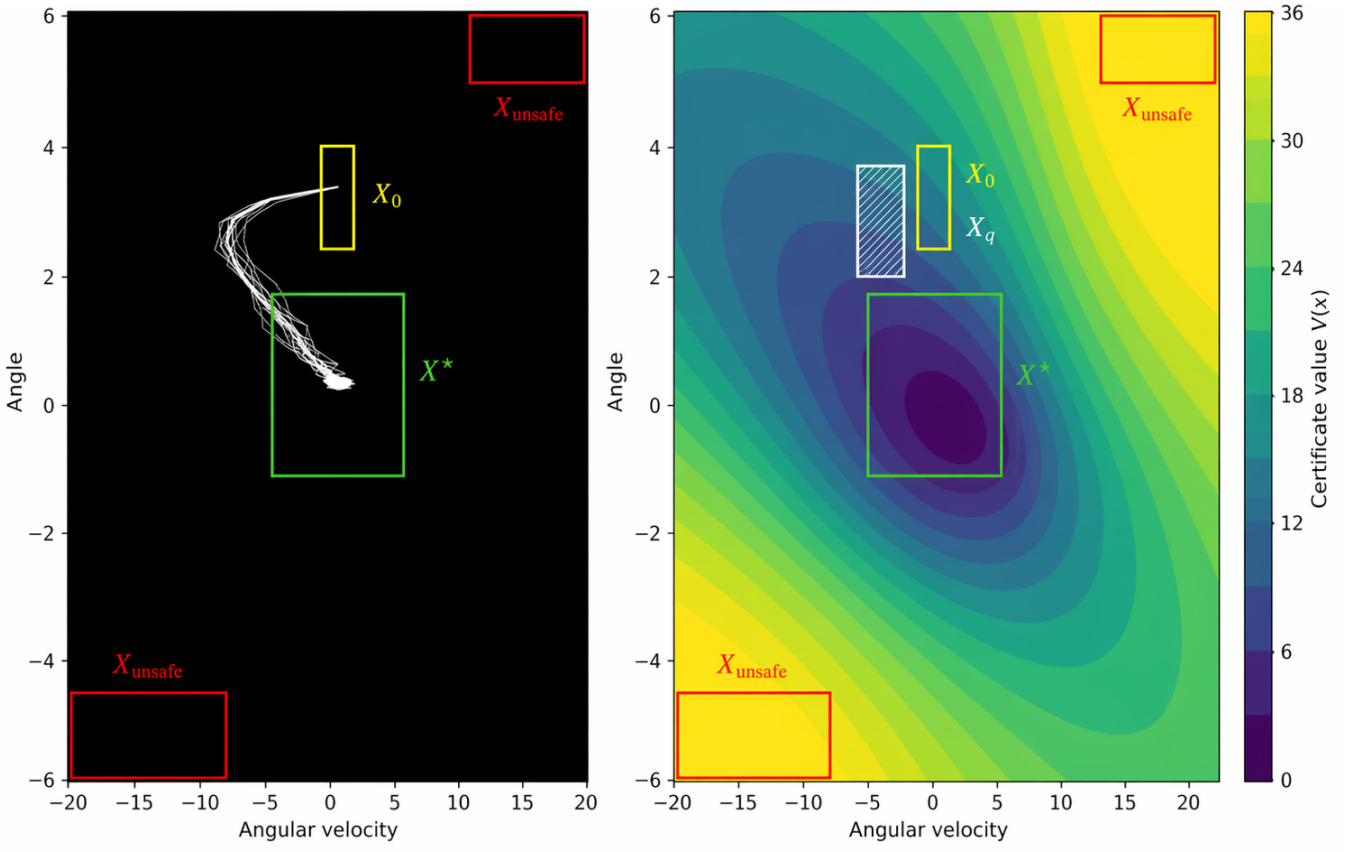

Figure 5: Conceptual illustration of the reach-avoid certificate viewpoint for the stochastic inverted-pendulum running example. The left panel shows representative closed-loop trajectories starting from the initial set X_0 and evolving toward the target set $X_\star$ while avoiding the unsafe regions $X_\oslash$, labelled in the plot as $X\ unsafe$. The right panel shows the corresponding certificate landscape V over the same state space. Lower certificate values correspond to states that are safer according to the certificate, while higher values correspond to states closer to violating the reach-avoid objective. The shaded region X_q illustrates a localised model change, which becomes relevant for the VeRecycle setting introduced later.

For a fixed certificate V, the original dynamics induce the generator $G_\pi V$, while the modified dynamics induce the generator $\tilde{G}_\pi V$. Since the drift and diffusion agree outside X_q, the two generators also agree outside X_q:

$$(\tilde{G}_\pi V)(x) = (G_\pi V)(x) \qquad \forall x \in X \setminus X_q. \tag{26}$$

Inside X_q, this equality need not hold. Therefore, the original generator decrease condition may fail only inside the changed region.

We say that a pair $(\tilde{f}_\pi, \tilde{g}_\pi)$ is an *admissible modification* if it satisfies the same regularity assumptions as the original dynamics, so that the modified stochastic differential equation is well posed, and if it coincides with the original dynamics outside X_q. Inside X_q, the modified drift and diffusion may differ arbitrarily subject to these conditions.

Running example. ***Stochastic inverted pendulum, continued.*** *For the pendulum, a localised model change means that the dynamics are no longer trusted in a restricted part of the angle–velocity state space. For example, the pendulum may experience a different friction coefficient, a different torque response, or a stronger stochastic disturbance only for certain ranges of ω and θ. This changed region can be represented by a subset $X_q \subseteq X$, for instance a rectangle in the (ω, θ)-plane.*

Outside X_q, the nominal drift and diffusion remain unchanged. Inside X_q, the modified drift and diffusion may differ. This is exactly the setting where certificate reuse becomes relevant: the original certificate may still contain useful global information, even though the dynamics are no longer guaranteed to correspond to the original model everywhere.

3.6 Reclaimed guarantees

We now formalise how guarantees derived from the certificate can be preserved under such localised changes.

Original certified guarantee. Since V is a continuous-time reach-avoid certificate with parameters (α, β, ζ), the certificate theorem of Neustroev et al. (2025, Theorem 1 and Corollary 1) yields the original certified reach-avoid bound

$$\mathbb{P}_{x_0}(X_\tau \in X_\star) \;\geq\; \rho := 1 - \frac{\alpha}{\beta}, \qquad \forall x_0 \in X_0. \tag{27}$$

This bound is the reference guarantee that we aim to reclaim after a localised model change.

When the modified dynamics are considered, X_t and τ are understood with respect to the modified SDE. Equivalently, one may write $\tilde{X}_t$ and $\tilde{\tau}$ for the modified process and stopping time.

Definition 6 (Universally valid reclaimed guarantee)**.** *Let V be a continuous-time reach–avoid certificate for the original closed-loop dynamics with parameters (α, β, ζ), and let $X_q \subseteq X$ be the region where the dynamics may change. A value $\rho_{\mathrm{rec}} \in [0, \rho]$ is called a* universally valid reclaimed guarantee *if, for every admissible modification $(\tilde{f}_\pi, \tilde{g}_\pi)$ supported on X_q, the same fixed certificate V certifies the reach–avoid bound*

$$\mathbb{P}_{x_0}^{(\tilde{f}_\pi, \tilde{g}_\pi)}(X_\tau \in X_\star) \geq \rho_{\mathrm{rec}} \qquad \forall x_0 \in X_0. \tag{28}$$

Here, "certifies" means that the continuous-time reach–avoid certificate theorem can be applied to V under the modified dynamics, with the same initial level α and with an unsafe level corresponding to the reclaimed bound.

Maximal reclaimed guarantee. In general, multiple reclaimed lower bounds may be certifiable by reusing V. We are interested in the strongest one:

$$\rho^\star_{\mathrm{rec}}(V) := \sup \{ r \in [0, \rho] \mid r \text{ is a universally valid reclaimed guarantee in the sense of definition 6} \}. \tag{29}$$

The trivial bound 0 is included, so the set in eq. (29) is nonempty.

Certificate infimum over the changed region. The key quantity governing the recovered guarantee is

$$m := \inf_{x \in X_q} V(x). \tag{30}$$

Intuitively, m is the lowest certificate level attained inside the changed region. It identifies whether X_q overlaps with low-certificate states that are important to the original proof, or whether the changed region already lies entirely above a higher certificate level.

Running example. ***Stochastic inverted pendulum, continued.*** *In the pendulum example, m is the lowest certificate value attained anywhere in the changed angle–velocity region. If X_q contains low-certificate states, then the original proof may rely on generator decrease in that region, and the reclaimed guarantee can become weak or even trivial. If X_q lies only in a high-certificate part of the landscape, then the proof can avoid relying on generator decrease inside that region, and a greater portion of the original guarantee can be recovered. This intuition is exactly what the next chapter turns into a formal reclaiming rule.*

Takeaway for the next section. The objects defined above reduce the reclaiming problem to a precise mathematical question: given a fixed reach–avoid certificate V and a changed region X_q, what is the largest guarantee that can still be certified uniformly over all admissible local changes by reusing V, without retraining the certificate? In Chapter 4, we answer this by deriving the continuous-time analogue of the discrete-time VeRecycle threshold. The key technical difference is that the discrete-time one-step expected decrease condition is replaced by the generator decrease condition from eq. (23). We define a candidate threshold $\rho_{\mathrm{rec}}(V)$ and prove that it equals the maximal reclaimed guarantee:

$$\rho^\star_{\mathrm{rec}}(V) = \rho_{\mathrm{rec}}(V). \tag{31}$$

4 Continuous-Time VeRecycle

This chapter answers RQ1. We prove that the VeRecycle reclaiming principle can be extended soundly to continuous-time stochastic differential equations with generator-based reach–avoid certificates. More precisely, we derive a continuous-time analogue of the VeRecycle reclaiming rule for the reach–avoid setting of section 3, and show that the maximal reclaimed guarantee is determined by the local certificate infimum

$$m := \inf_{x \in X_q} V(x).$$

Setup. We study how much of the original certified bound can still be reclaimed when the system dynamics change locally inside X_q, while the certificate V itself is kept fixed.

Throughout this section, we assume:

- a compact state space X,
- original dynamics (f_π, g_π),
- modified dynamics $(\tilde{f}_\pi, \tilde{g}_\pi)$ supported on X_q,
- a reach–avoid certificate V with parameters (α, β, ζ),
- a change region satisfying
$$X_q \cap X_\star = \emptyset,$$
- and
$$m := \inf_{x \in X_q} V(x).$$

The assumption $X_q \cap X_\star = \emptyset$ ensures that states in the changed region are not excluded from the certificate decrease condition by belonging to the target set. No disjointness from $X_\oslash$ is required for the argument below.

Admissible modifications. An *admissible modification supported on* X_q is any pair $(\tilde{f}_\pi, \tilde{g}_\pi)$ such that:

- $(\tilde{f}_\pi, \tilde{g}_\pi)$ satisfies the same regularity assumptions as the original dynamics, so that the modified SDE is well posed;
- $\tilde{f}_\pi(x) = f_\pi(x)$ and $\tilde{g}_\pi(x) = g_\pi(x)$ for all $x \in X \setminus X_q$.

Inside X_q, the modified drift and diffusion may differ arbitrarily subject to these admissibility conditions.

Original certified bound. Since V is a reach–avoid certificate with parameters (α, β, ζ), we have

$$V(x) \le \alpha \qquad \forall x \in X_0, \qquad V(x) \ge \beta \qquad \forall x \in X_\oslash,$$

and $0 < \alpha < \beta$. By eq. (27), the corresponding original certified reach–avoid bound is

$$\rho = 1 - \frac{\alpha}{\beta}.$$

Proof rule. The key technical tool is the continuous-time certificate framework of Neustroev et al. (2025, Theorem 1 and Corollary 1). Their main theorem is formulated for reach-stay-avoid specifications, while Corollary 1 shows that the reach–avoid part can be certified on its own by omitting the stay conditions. In the time-homogeneous reach–avoid setting used here, this yields

$$\mathbb{P}_{x_0}(X_\tau \in X_\star) \ge 1 - \frac{\alpha}{\beta}.$$

How it is used. Our argument uses Neustroev et al. (2025, Theorem 1 and Corollary 1) as a black-box proof rule. We do not reprove the continuous-time certificate theorem itself. Instead, we show that after a suitable threshold adjustment, the reused certificate V still satisfies the hypotheses of that theorem for the modified dynamics.

Why reclaiming is needed. Recall that the certificate V induces the guarantee $1-\alpha/\beta$ under the original dynamics. However, when the dynamics are modified inside X_q, the certificate may no longer satisfy the generator decrease condition there. As a result, the original guarantee is not automatically preserved under the modified dynamics.

Local quantity controlling reclaiming. The key idea of VeRecycle is to reuse the same certificate V and adjust the guarantee according to how the certificate behaves on the changed region. Intuitively, if X_q contains low-certificate states, then the original proof may rely on the generator decrease condition in the changed region. Since the modified dynamics may violate that decrease condition inside X_q, only a weaker guarantee may remain certifiable. Conversely, if all states in X_q already have high certificate values, then the proof can avoid relying on generator decrease inside the changed region, and a greater portion of the original guarantee can be preserved.

This behaviour is captured by the quantity m from eq. (30), which represents the worst-case certificate value inside the changed region.

The central question is therefore whether the same infimum-based reclaiming structure also survives in continuous time under the generator-based proof rule.

Reading guide. *The argument proceeds in three steps. First, eqs.* (32) *and* (33) *define the candidate reclaimed threshold* $\rho_{\mathrm{rec}}(V)$*. Second, Lemma 1 proves that this threshold is valid for every admissible localised change. Third, Lemma 2 proves that no larger threshold in* $[0, \rho]$ *can be certified from the same certificate* V *through the proof rule of Neustroev et al. (2025, Theorem 1 and Corollary 1). Readers mainly interested in the final result may focus on Definition 7, Theorem 1, and the cases* $m \le \alpha$*,* $\alpha < m < \beta$*, and* $m \ge \beta$*.*

4.1 Reclaimed threshold

Objective. The VeRecycle objective is to determine the largest universally valid reclaimed guarantee, in the sense of definition 6, that can be obtained by reusing the fixed certificate V after an arbitrary admissible localised change on X_q.

Two controlling quantities. The analysis below shows that two independent constraints control any such bound: the original certified threshold ρ, and a second constraint induced by the minimal value of V on the changed region, m; see Lemmas 1 and 2.

Definition 7 (Candidate reclaimed threshold induced by V)**.** *Let* V *be a reach–avoid certificate, and let* m *be as defined in eq.* (30)*. The* candidate reclaimed threshold induced by V *is*

$$\rho_{\mathrm{rec}}(V) := \begin{cases} 0, & m \le \alpha, \\ 1 - \dfrac{\alpha}{\min(\beta, m)}, & m > \alpha. \end{cases} \tag{32}$$

For $m > 0$*, this can equivalently be written as*

$$\rho_{\mathrm{rec}}(V) = \min\Big(\rho, \max\Big(0,\, 1 - \frac{\alpha}{m}\Big)\Big)\,. \tag{33}$$

The other definition in eq. (32) *includes the degenerate case* $m = 0$*.*

The formula has three regimes: if $m \le \alpha$, no positive bound can be reclaimed; if $\alpha < m < \beta$, the reclaimed bound increases with m; and if $m \ge \beta$, the original certified bound is fully recovered.

Role of the two forms. The piecewise form in eq. (32) is used in the proof because it separates the cases $m \le \alpha$ and $m > \alpha$. The equivalent form in eq. (33) makes the connection with the discrete-time VeRecycle rule more explicit: the reclaimed threshold is the original certified bound ρ, capped by the additional constraint induced by the changed-region infimum m.

Proof structure. The formula in eq. (32) is first treated as a candidate reclaimed threshold. Lemma 1 shows that this candidate is valid for every admissible modification supported on X_q. Lemma 2 then shows that no larger threshold in $[0, \rho]$ can be certified using the same certificate V through the proof rule of Neustroev et al. (2025, Theorem 1 and Corollary 1). Together, the two lemmas establish that $\rho_{\mathrm{rec}}(V)$ is the maximal reclaimed guarantee obtainable by reusing V.

4.2 Lemma 1: Lower bound

Proof intuition. Lemma 1 proves the first direction: the candidate threshold $\rho_{\mathrm{rec}}(V)$ in eq. (32) is always reclaimable. Equivalently, it provides a lower bound on the maximal reclaimed guarantee $\rho^{\star}_{\mathrm{rec}}(V)$.

The certificate V is kept fixed after the dynamics change. The only difficulty is that, inside the changed region X_q, the modified dynamics may no longer satisfy the original generator decrease condition. We therefore lower the unsafe threshold from β to

$$\beta' := \min(\beta, m).$$

The key observation is that any state with $V(x) < \beta'$ cannot lie in X_q, because every state in X_q has certificate value at least $m \ge \beta'$. Hence, on all states where decrease must be checked, the original and modified generators coincide. Therefore, the original certificate theorem can be reused directly to prove the formal lower bound stated in Lemma 1.

Lemma 1 (Lower bound)**.** *For every admissible modification* $(\tilde{f}_\pi, \tilde{g}_\pi)$ *supported on* X_q *and every* $x_0 \in X_0$*,*

$$\mathbb{P}_{x_0}^{(\tilde{f}_\pi, \tilde{g}_\pi)}(X_\tau \in X_\star) \ge \rho_{\mathrm{rec}}(V).$$

Proof. Fix an admissible modification $(\tilde{f}_\pi, \tilde{g}_\pi)$ supported on X_q and an initial state $x_0 \in X_0$. Define

$$\beta' := \min(\beta, m). \tag{34}$$

If $\beta' \le \alpha$, then $\rho_{\mathrm{rec}}(V) = 0$ by eq. (32), and the claim is immediate since probabilities are nonnegative. It therefore remains to consider the case

$$0 < \alpha < \beta'.$$

We verify that V satisfies the reach–avoid certificate conditions for the modified dynamics with the same initial set X_0, target set $X_\star$, unsafe set $X_\oslash$, initial threshold α, and unsafe threshold β'.

1. *Initial condition.* Since V is a reach–avoid certificate for the original system,
$$V(x) \le \alpha \qquad \forall x \in X_0.$$

2. *Unsafe condition.* Again by the original certificate property,
$$V(x) \ge \beta \qquad \forall x \in X_\oslash.$$
Since $\beta' = \min(\beta, m) \le \beta$, it follows that
$$V(x) \ge \beta' \qquad \forall x \in X_\oslash.$$

3. *Generator condition.* Let $x \in X \setminus X_\star$ satisfy
$$V(x) < \beta'.$$
Since $\beta' \le m$, we obtain
$$V(x) < m.$$
By the definition of m in eq. (30), no point in X_q can have certificate value strictly smaller than m. Hence
$$x \notin X_q.$$
By eq. (25), the original and modified drift and diffusion coincide outside X_q. Therefore, the corresponding generators also coincide there:
$$\tilde{G}_\pi V(x) = G_\pi V(x).$$
Moreover, because $V(x) < \beta' \le \beta$, the original certificate condition implies
$$G_\pi V(x) \le -\zeta.$$
Therefore,
$$\tilde{G}_\pi V(x) \le -\zeta \qquad \text{for all } x \in X \setminus X_\star \text{ such that } V(x) < \beta'.$$

Thus, all certificate conditions hold for the modified dynamics with thresholds $0 < \alpha < \beta'$. By Neustroev et al. (2025, Theorem 1 and Corollary 1),
$$\mathbb{P}_{x_0}^{(\tilde{f}_\pi, \tilde{g}_\pi)}(X_\tau \in X_\star) \ge 1 - \frac{\alpha}{\beta'}.$$
Substituting eq. (34), this gives
$$\mathbb{P}_{x_0}^{(\tilde{f}_\pi, \tilde{g}_\pi)}(X_\tau \in X_\star) \ge 1 - \frac{\alpha}{\min(\beta, m)}.$$
By eq. (32), this is exactly $\rho_{\mathrm{rec}}(V)$. □

4.3 Lemma 2: Upper bound

Proof intuition. We now show that the lower bound from Lemma 1 is optimal among all reclaimed bounds in $[0, \rho]$ that can be certified by reusing the same certificate V through the proof rule of Neustroev et al. (2025, Theorem 1 and Corollary 1). Any lower bound obtained in this way must be of the form
$$\max\left(0,\, 1 - \frac{\alpha}{\beta'}\right)$$
for some threshold $\beta' > 0$.

Two constraints limit how large such a bound can be. First, reclaiming is defined relative to the original certified bound, so a reclaimed threshold cannot exceed ρ. Second, because modifications inside X_q are arbitrary, one can choose an admissible modification that makes any generator decrease impossible at a state in X_q. Therefore, if a theorem application is to remain valid for all admissible modifications, it cannot rely on the generator condition inside X_q; this forces $\beta' \le m$. Combining these two constraints yields the claimed upper bound.

Lemma 2 (Upper bound)**.** *Let $\rho' \in [0, \rho]$ be such that for every admissible modification $(\tilde{f}_\pi, \tilde{g}_\pi)$ supported on X_q and every $x_0 \in X_0$, the bound*
$$\mathbb{P}_{x_0}^{(\tilde{f}_\pi, \tilde{g}_\pi)}(X_\tau \in X_\star) \ge \rho'$$
can be obtained by applying Neustroev et al. (2025, Theorem 1 and Corollary 1) with the same certificate function V (possibly with a different unsafe threshold). Then
$$\rho' \le \rho_{\mathrm{rec}}(V).$$

Proof. Fix ρ' as in the statement. Any lower bound obtained by applying Neustroev et al. (2025, Theorem 1 and Corollary 1) with the certificate function V must be of the form

$$\max\left(0,\, 1 - \frac{\alpha}{\beta'}\right)$$

for some $\beta' > 0$ such that the certificate conditions hold with threshold β'.

If $\beta' \leq \alpha$, then the resulting lower bound is 0, and the claim is immediate. It therefore suffices to consider the case

$$0 < \alpha < \beta'.$$

1. *First constraint.* Since ρ' is a reclaimed threshold, by definition it cannot exceed the original certified bound ρ. Hence

$$\rho' \leq \rho. \tag{35}$$

2. *Second constraint.* Suppose, for contradiction, that

$$\beta' > m.$$

Then, by the definition of $m = \inf_{x \in X_q} V(x)$, there exists $x^\dagger \in X_q$ such that

$$V(x^\dagger) < \beta'.$$

Because admissible modifications may alter the dynamics inside X_q arbitrarily subject to the admissibility conditions, choose one such modification satisfying

$$\tilde{f}_\pi(x^\dagger) = 0, \qquad \tilde{g}_\pi(x^\dagger) = 0.$$

At this point, the modified generator satisfies

$$(\tilde{G}_\pi V)(x^\dagger) = 0.$$

Since $X_q \cap X_\star = \emptyset$, we have $x^\dagger \in X \setminus X_\star$. Moreover, $V(x^\dagger) < \beta'$. Therefore, if Neustroev et al. (2025, Theorem 1 and Corollary 1) is to be applied with threshold β', the decrease condition must hold at $x^\dagger$, which would require

$$(\tilde{G}_\pi V)(x^\dagger) \leq -\zeta.$$

But this is impossible, because $(\tilde{G}_\pi V)(x^\dagger) = 0$ and $\zeta > 0$. Hence no theorem application that is valid uniformly over all admissible modifications can permit a state in X_q with certificate value below β'. Therefore,

$$V(x) \geq \beta' \qquad \forall x \in X_q.$$

It follows that

$$\beta' \leq \inf_{x \in X_q} V(x) = m.$$

If $m \leq \alpha$, this contradicts the case assumption $0 < \alpha < \beta'$. Hence no positive reclaimed threshold can be certified in this case, and the only possible certified lower bound is the trivial bound $0 = \rho_{\mathrm{rec}}(V)$.

We have already handled the case $m \leq \alpha$. Therefore, the only remaining case is

$$m > \alpha.$$

In this case, the nontrivial branch of eq. (32) applies. Since $\beta' \leq m$, any bound certified with threshold β' is at most

$$\rho' \leq 1 - \frac{\alpha}{m}. \tag{36}$$

Combining eqs. (35) and (36) gives

$$\rho' \leq \min\left(\rho, 1 - \frac{\alpha}{m}\right). \tag{37}$$

For $m > \alpha$, the expression on the right-hand side is exactly the equivalent form of $\rho_{\mathrm{rec}}(V)$ in eq. (33). Therefore,

$$\rho' \leq \rho_{\mathrm{rec}}(V).$$

□

4.4 Main theorem

Meaning of the result. The two lemmas together show that $\rho_{\mathrm{rec}}(V)$ is not only a valid reclaimed threshold, but the strongest one that can be certified from the fixed certificate V through the proof rule of Neustroev et al. (2025, Theorem 1 and Corollary 1).

Theorem 1 (Continuous-time VeRecycle)**.** *Let V be a continuous-time reach–avoid certificate with parameters (α, β, ζ), and let*

$$m := \inf_{x \in X_q} V(x).$$

Let

$$\rho := 1 - \frac{\alpha}{\beta}$$

be the original certified reach–avoid bound.

Then

$$\rho_{\mathrm{rec}}(V) = \begin{cases} 0, & m \le \alpha, \\ 1 - \dfrac{\alpha}{\min(\beta, m)}, & m > \alpha \end{cases} \tag{38}$$

is the maximal reclaimed reach–avoid threshold in $[0, \rho]$ that can be certified by reusing the certificate V through the reach–avoid certificate theorem of Neustroev et al. (2025, Theorem 1 and Corollary 1), uniformly over all admissible modifications supported on X_q. Equivalently,

$$\rho^{\star}_{\mathrm{rec}}(V) = \rho_{\mathrm{rec}}(V).$$

Proof. Lemma 1 shows that $\rho_{\mathrm{rec}}(V)$ is a valid reclaimed threshold obtainable from the certificate V. Lemma 2 shows that no larger reclaimed threshold in $[0, \rho]$ can be certified through the proof rule of Neustroev et al. (2025, Theorem 1 and Corollary 1) while reusing the same certificate V. Hence $\rho_{\mathrm{rec}}(V)$ is maximal among all reclaimed thresholds that can be certified by reusing V through that proof rule. □

4.5 Chapter summary

This chapter established the continuous-time VeRecycle reclaiming rule. Once the original certificate V is fixed, the maximal reclaimed guarantee depends on the changed region X_q only through the certificate infimum $m = \inf_{x \in X_q} V(x)$. The proof uses the piecewise form in eq. (32), while the equivalent form in eq. (33) makes the connection to the discrete-time VeRecycle rule explicit. In the next section, we evaluate this rule empirically and examine whether the implementation reproduces the predicted dependence of the reclaimed guarantee on m.

5 Evaluation

This chapter answers RQ2 by evaluating continuous-time VeRecycle on the stochastic inverted-pendulum benchmark introduced in Chapter 2. The evaluation starts from an already-certified closed-loop system: a fixed neural policy π and a neural reach–avoid certificate V. VeRecycle does not retrain either network. Instead, it reuses the certified pair (π, V) and computes how much of the original guarantee can still be recovered after a localised model change.

The experiments study three aspects of this reuse mechanism. First, we check whether the implemented guarantees follow the theoretical dependence on the changed-region certificate minimum $m = \inf_{x \in X_q} V(x)$, using a certified IBP lower bound $m_{\mathrm{lb}} \leq m$ in the implementation. Second, we compare VeRecycle with full re-certification to measure the computational benefit of reusing the fixed certificate. Third, we compare continuous-time VeRecycle with a naive discretise-then-reclaim workflow to illustrate how coarse sampling can miss continuous changed regions.

The evaluation continues as follows. Section 5.1 describes the benchmark, changed regions, VeRecycle computation, re-certification baseline, and reproducibility setup. The following subsections analyse the dependence on m_{lb}, compare representative scenarios, discuss runtime, and study the naive discretisation baseline.

5.1 Experimental setup

Benchmark. We consider the continuous-time stochastic inverted-pendulum benchmark used in the underlying certificate framework of Neustroev et al. (2025). The state is

$$x = (\omega, \theta),$$

where ω is the angular velocity and θ is the pendulum angle. In the notation of Neustroev et al. (2025), the angular velocity coordinate is denoted by $\phi = \dot{\theta}$; here we write ω to be consistent with the rest of the thesis.

The deterministic inverted-pendulum dynamics are

$$\ddot{\theta} = \frac{g}{L} \sin\theta + \frac{Mu - b\dot{\theta}}{m_{\mathrm{p}} L^2}.$$

The benchmark constants reported by Neustroev et al. (2025) are

$$g = 9.81, \qquad L = 0.5, \qquad m_{\mathrm{p}} = 0.15, \qquad b = 0.1, \qquad M = 6,$$

with control bound $|u| \leq 1$. Here, g is gravitational acceleration, L is the pendulum length, m_{p} is the pendulum mass, b is the friction coefficient, and M is the torque scaling constant. We write m_{p} for the pendulum mass to avoid confusion with the certificate minimum $m = \inf_{x \in X_q} V(x)$ used later.

In the stochastic benchmark, the angular-velocity equation is disturbed by a Wiener process with noise scale σ. With the fixed neural policy $u_t = \pi(\omega_t, \theta_t)$, we write the controlled dynamics as

$$\begin{aligned} d\omega_t &= \left(\frac{g}{L} \sin(\theta_t) + \frac{M\pi(\omega_t, \theta_t) - b\omega_t}{m_{\mathrm{p}} L^2} \right) dt + \sigma \, dW_t, \\ d\theta_t &= \omega_t \, dt. \end{aligned}$$

Equivalently, the closed-loop drift and diffusion are

$$f_\pi(\omega, \theta) = \begin{bmatrix} \dfrac{g}{L} \sin\theta + \dfrac{M\pi(\omega, \theta) - b\omega}{m_{\mathrm{p}} L^2} \\ \omega \end{bmatrix}, \qquad g_\pi(\omega, \theta) = \begin{bmatrix} \sigma \\ 0 \end{bmatrix}.$$

In the repository implementation used for this thesis, the noise scale is $\sigma = 2.0$.

The state space and specification sets are

$$\begin{aligned} X &= [-20, 20] \times [-2\pi, 2\pi], \\ X_0 &= [-1, 1] \times \left[\frac{3\pi}{4}, \frac{5\pi}{4} \right], \\ X_\star &= [-4, 4] \times \left[-\frac{\pi}{2}, \frac{\pi}{2} \right], \end{aligned}$$

and

$$X_\oslash = \left([-20, -10] \times \left[-2\pi, -\frac{3\pi}{2} \right] \right) \cup \left([10, 20] \times \left[\frac{3\pi}{2}, 2\pi \right] \right).$$

The original benchmark uses a reach-avoid-stay specification with $\varepsilon = 0.9$ and $\delta = 0.9$. In this evaluation, we use the reach-avoid part of the same benchmark sets, with original certified reach–avoid probability 0.9.

The original certified reach–avoid guarantee has the form

$$1 - \frac{\alpha}{\beta},$$

where α upper-bounds the certificate on the initial set and β lower-bounds it on the unsafe set (Neustroev et al., 2025). Since multiplying V, α, and β by the same positive constant does not change this ratio, the certified probability can be written in the normalised scale

$$\alpha = 1, \qquad \beta = 10,$$

so that

$$1 - \frac{\alpha}{\beta} = 1 - \frac{1}{10} = 0.9.$$

However, the implementation and all numerical tables use the raw checkpoint scale. In the final run this scale gives

$$\alpha = 1.8103, \qquad \beta = 18.1032,$$

and the reported values of m_{lb} are in this same raw scale.

The fixed policy is represented by a neural network with two hidden layers of 64 tanh units. The certificate is represented by a neural network with two hidden layers of 32 tanh units, followed by a softplus activation to enforce nonnegativity. Verification follows the same IBP-based framework as the underlying continuous-time certificate pipeline of Neustroev et al. (2025). In our implementation, we use a batch size of 256 and a verifier mesh size of 400, as in the pendulum setting of Neustroev et al. (2025), but set the maximum verification depth to 4.

The final certificate used by VeRecycle is the network stored in `certified.pt`. This checkpoint is fixed during all reclaiming experiments. The training evolution of such a certificate was illustrated earlier in Figure 3, but that figure is not part of the evaluation, it is included only to explain the neural-certificate workflow. All VeRecycle guarantees reported below use only this final fixed certificate V.

Changed regions. Starting from the certified base system, we introduce localised regions X_q in which the dynamics may differ from the base model. There are two separate choices in each modified experiment. First, we choose the location of the changed region

$$X_q = [\omega_{\min}, \omega_{\max}] \times [\theta_{\min}, \theta_{\max}],$$

which determines the local certificate value

$$m = \inf_{x \in X_q} V(x).$$

Second, we choose the type of local dynamics modification applied inside X_q. Outside X_q, the dynamics are unchanged.

Let $\mathbf{1}_{X_q}(x)$ be the indicator of the changed region. The original closed-loop dynamics are

$$dX_t = f_\pi(X_t)\, dt + g_\pi(X_t)\, dW_t.$$

For a drift-bias change with vector $c = (c_\omega, c_\theta)$, we use

$$\tilde{f}_\pi(x) = f_\pi(x) + \mathbf{1}_{X_q}(x)c, \qquad \tilde{g}_\pi(x) = g_\pi(x).$$

For a diffusion-scale change with scale s, we use

$$\tilde{f}_\pi(x) = f_\pi(x), \qquad \tilde{g}_\pi(x) = \big(1 + (s-1)\mathbf{1}_{X_q}(x)\big)\, g_\pi(x).$$

For the absorbing stress-test change, we use

$$\tilde{f}_\pi(x) = \big(1 - \mathbf{1}_{X_q}(x)\big)\, f_\pi(x), \qquad \tilde{g}_\pi(x) = \big(1 - \mathbf{1}_{X_q}(x)\big)\, g_\pi(x).$$

The indicator notation is used to describe the intended local support of each modification. Strictly speaking, a discontinuous indicator can violate the regularity assumptions used in the theoretical SDE model at the boundary of X_q. This does not affect the VeRecycle computation itself, which only uses the support set X_q and the lower bound on V over that set. If exact regularity is required, the indicator can be replaced by a smooth bump function that equals one on the interior of X_q, decays smoothly near the boundary, and is zero outside a slightly enlarged local region. The experiments should therefore be read as localised perturbation scenarios used to evaluate the reclaiming mechanism, rather than as a claim that the discontinuous indicator form is the only admissible continuous-time modification. Thus, the experiments change the certified system in two ways: by moving the changed set X_q to different parts of the state space, and by changing the local dynamics type inside that set.

These regions model situations in which only part of the system changes while the policy and the original certificate remain fixed. The regions X_q are selected to explore how the reclaimed guarantee depends on the local certificate value. Concretely, we use changed regions with different positions and sizes, thereby inducing different values of

$$m = \inf_{x \in X_q} V(x).$$

Table 1: Changed-region locations and local dynamics modifications. Coordinates are in the state order (ω, θ). Outside X_q, the original dynamics are unchanged.

Scenario	X_q	Type	Parameter
Low certificate margin	$[4.0, 7.0] \times [2.0, 3.5]$	diffusion	$s = 1.5$
Borderline margin	$[4.5, 6.0] \times [2.1, 3.0]$	diffusion	$s = 2.0$
Mid transition	$[7.0, 8.0] \times [2.0, 3.0]$	diffusion	$s = 2.0$
High transition	$[8.0, 9.5] \times [2.5, 3.5]$	diffusion	$s = 1.5$
Near original bound	$[11.0, 12.5] \times [3.5, 4.5]$	diffusion	$s = 1.1$
Mild drift band	$[5.0, 8.0] \times [2.0, 3.5]$	drift	$c = (-0.8, -0.4)$
Strong drift corridor	$[7.5, 10.5] \times [2.0, 4.0]$	drift	$c = (-1.5, -0.8)$
Central absorbing	$[1.5, 4.5] \times [1.0, 2.6]$	absorbing	–
Near-target absorbing	$[2.5, 5.5] \times [0.8, 2.0]$	absorbing	–
Near-initial absorbing	$[-1.5, 1.5] \times [2.0, 3.4]$	absorbing	–
Low-margin absorbing	$[4.0, 7.0] \times [2.0, 3.5]$	absorbing	–
Transition absorbing	$[7.0, 8.0] \times [2.0, 3.0]$	absorbing	–
Distant high-margin absorbing	$[-12.5, -10.5] \times [-4.5, -3.0]$	absorbing	–
Far high-margin absorbing	$[11.0, 12.5] \times [3.5, 4.5]$	absorbing	–

This allows validating the implementation against the theoretical dependence of ρ_{rec} on m. The changed regions are fixed axis-aligned rectangles with predetermined positions and sizes. We include regions along typical closed-loop trajectories, regions in high-certificate parts of the state space, transition regions, drift-modified regions, and absorbing stress-test regions. This collection produces low, intermediate, and high values of the local certificate minimum.

The chosen regions are not intended to exhaust all possible perturbations of the pendulum dynamics. Rather, they are selected to cover the certificate regimes that determine the theoretical reclaiming rule:

$$m \le \alpha, \qquad \alpha < m < \beta, \qquad m \ge \beta.$$

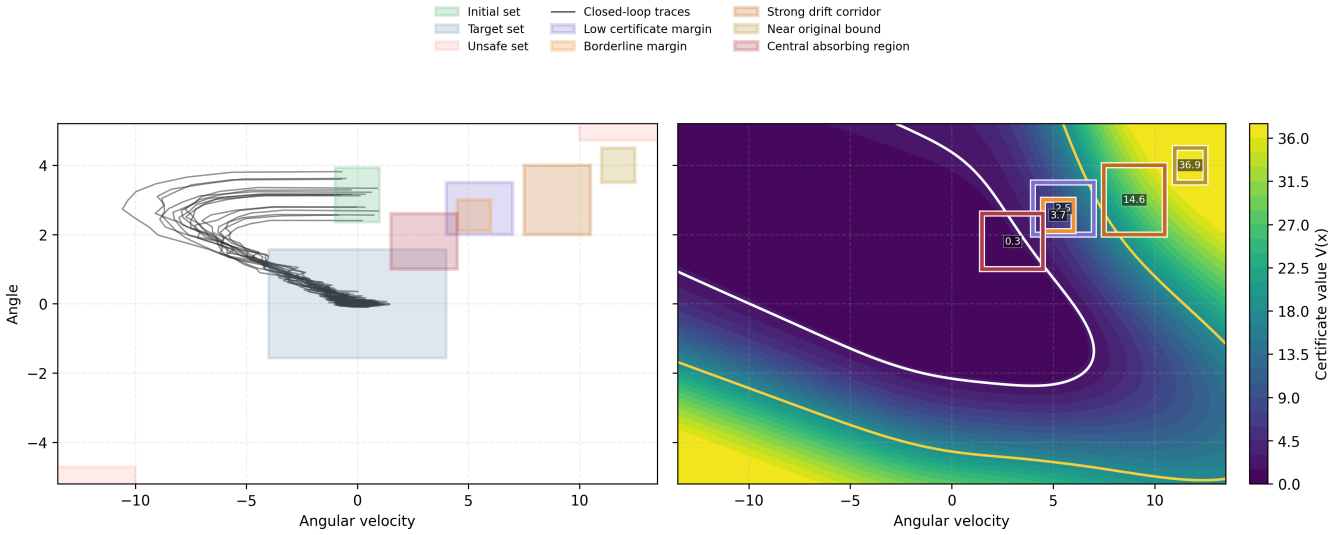

Figure 6: State-space layout for the stochastic inverted-pendulum evaluation. Panel (a) shows the benchmark geometry, representative closed-loop traces, and the initial, target, unsafe, and changed regions. Panel (b) shows the certificate landscape V, together with representative changed regions X_q and their local certificate lower bounds. The selected regions cover low-, intermediate-, and high-certificate regimes, which determine how much of the original guarantee can be reclaimed.

This makes the experiment representative for the mechanism being tested, because continuous-time VeRecycle depends on a changed region only through the certificate lower bound over that region. For other systems or tasks, the same methodology applies: select changed sets that probe the relevant low-, intermediate-, and high-certificate parts of the state space. The computational difficulty of computing tight lower bounds may increase with dimension, but the reclaiming rule itself is independent of the particular geometry used in this two-dimensional benchmark.

The state-space layout, representative changed regions, and certificate landscape are shown in Figure 6. The labels in that figure indicate where the representative scenarios used later in this section lie in state space: the low-margin and borderline regions sit near low certificate values, the strong-drift corridor lies farther along the closed-loop flow, and the near-original-bound region lies in a high-certificate part of the state space. The reported region locations and sizes are fixed for all evaluation runs, making the experiments reproducible and supporting a controlled study of how the reclaimed guarantee depends on the local certificate minimum.

VeRecycle computation. For each changed region, we compute a sound interval-bound-propagation (IBP) lower bound

$$m_{\mathrm{lb}} \leq \inf_{x \in X_q} V(x),$$

which is a conservative lower bound on the worst-case certificate value over the modified region. The bound is computed using the IBP-based verification pipeline of Neustroev et al. (2025), applied over a fixed mesh of rectangular subsets. This makes the computation efficient while preserving soundness. The resulting value m_{lb} is then substituted into the continuous-time VeRecycle reclaiming rule to obtain the certified guarantee

$$\rho_{\mathrm{rec}} = \begin{cases} 0, & m_{\mathrm{lb}} \leq \alpha, \\ 1 - \dfrac{\alpha}{\min(\beta, m_{\mathrm{lb}})}, & m_{\mathrm{lb}} > \alpha. \end{cases}$$

Since $m_{\mathrm{lb}} \leq \inf_{x \in X_q} V(x)$ and the reclaiming rule is monotone in the local certificate minimum, using m_{lb} can only make the reported guarantee more conservative.

The IBP parameters are kept fixed across the main reclaiming experiments. The local VeRecycle lower-bound computation partitions only the changed region X_q; in the final reclaiming runs, this local computation uses rectangular cells of width 0.05 and an IBP batch size of 1000. The local mesh controls a tightness–runtime trade-off: a finer partition can tighten the lower bound m_{lb}, while a coarser partition may make the bound more conservative. Since the reclaiming rule is monotone in m_{lb}, this conservatism can only reduce ρ_{rec} and cannot lead to over-reclaiming.

Additional mesh-refinement checks with cell widths of 0.05, 0.02, 0.01, and 0.005 produced the same qualitative recovery regime, with runtime increasing as the mesh was refined. Within higher-dimensional systems, a uniform rectangular mesh may perform inadequately because the number of cells grows rapidly with dimension; effective implementations would therefore use coarser local partitions, adaptive splitting, or stronger relaxation methods to focus computation on the parts of X_q that determine the lower bound.

Re-certification baseline. We compare VeRecycle against re-certification. In the re-certification baseline, the certificate is retrained for the modified system and then verified using the standard neural certificate pipeline. The re-certification baseline uses the same verifier mesh size as the original certificate-verification configuration, with a batch size of 128 and a maximum verification depth of 4.

The re-certification run is configured with target reach–avoid probability

$$p_{\mathrm{recert}} = 0.9.$$

If re-certification returns a valid certificate, we report the certified target probability:

$$\rho_{\mathrm{recert}} = p_{\mathrm{recert}} = 0.9.$$

If no valid certificate is obtained within the time budget, or if the modified dynamics are unsupported by the re-certification implementation, we report

$$\rho_{\mathrm{recert}} = 0.$$

Thus, the comparison is between two certified workflows: VeRecycle reuses the existing certificate and computes a local certified bound from m_{lb}, whereas re-certification attempts to construct and verify a new certificate for the modified system. This comparison is intentionally asymmetric: VeRecycle returns the certified bound obtainable from the fixed original certificate and the sound lower bound m_{lb}, whereas re-certification is evaluated as a success-or-failure attempt to certify a fixed target probability. Sampled certificate-level ratios are not used as guarantees in the comparison.

Implementation and reproducibility. All reported VeRecycle results are computed from the fixed checkpoint `certified.pt`; no network weights are updated during reclaiming. The changed-region definitions, IBP verifier parameters, re-certification target probability, and plotting scripts are fixed across runs. The experiments are therefore reproducible from the checkpoint, the policy network, the changed-region specifications, and the verification configuration described above.

The code, trained checkpoints, changed-region specifications, and scripts used to generate the reported tables and figures are publicly available at https://github.com/sgrad20/Verecycle-Neural-Continuous-Time-Certificates, with the VeRecycle evaluation scripts and outputs stored under the `experiments_ct` and `experiments_ct_verecycle` directories. Empirical rollouts are used only for diagnostic comparison and are not used as certified guarantees.

Hardware and software environment. All runtime measurements are obtained with the same CPU-based Python implementation, using Python 3.11.7 and the dependencies specified in the experimental repository linked above. The absolute wall-clock times are therefore hardware-dependent, but the comparison is controlled because VeRecycle and re-certification are measured in the same software and execution environment.

All experiments were conducted on an ASUS ROG Strix G512LU laptop with an Intel Core i7-10750H CPU at 2.60 GHz, 16 GB RAM, and Microsoft Windows 11 Pro. The system also contains Intel UHD Graphics and an NVIDIA GeForce GTX 1660 Ti GPU; however, GPU acceleration was not used in the reported experiments because the implementation was configured to run on the CPU, and the Python environment used a CPU-only version of PyTorch.

5.2 Dependence on the local certificate minimum

This experiment illustrates the theoretical characterisation from RQ1 by checking whether the implementation follows the predicted dependence on the certificate infimum over the changed region.

The theoretical characterisation derived in Section 4 shows that the reclaimed guarantee depends on

$$m := \inf_{x \in X_q} V(x)$$

through

$$\rho_{\mathrm{rec}} = \min\left(1 - \frac{\alpha}{\beta}, \max\left(0, 1 - \frac{\alpha}{m}\right)\right).$$

Equivalently, the reclaimed guarantee is determined by the position of the local certificate minimum relative to the initial-set level α and the unsafe-set level β. If $m \leq \alpha$, VeRecycle cannot recover a positive guarantee. If $\alpha < m < \beta$, the guarantee increases as m increases. If $m \geq \beta$, the guarantee saturates at the original bound.

Figure 7 shows this dependence directly. The solid curve is the theoretical reclaiming rule, while the markers correspond to the experimental cases using the certified lower bound m_{lb}. The scenarios lie on the theoretical curve: low-certificate regions yield small or zero guarantees, intermediate regions yield partial recovery, and high-certificate regions recover the original bound.

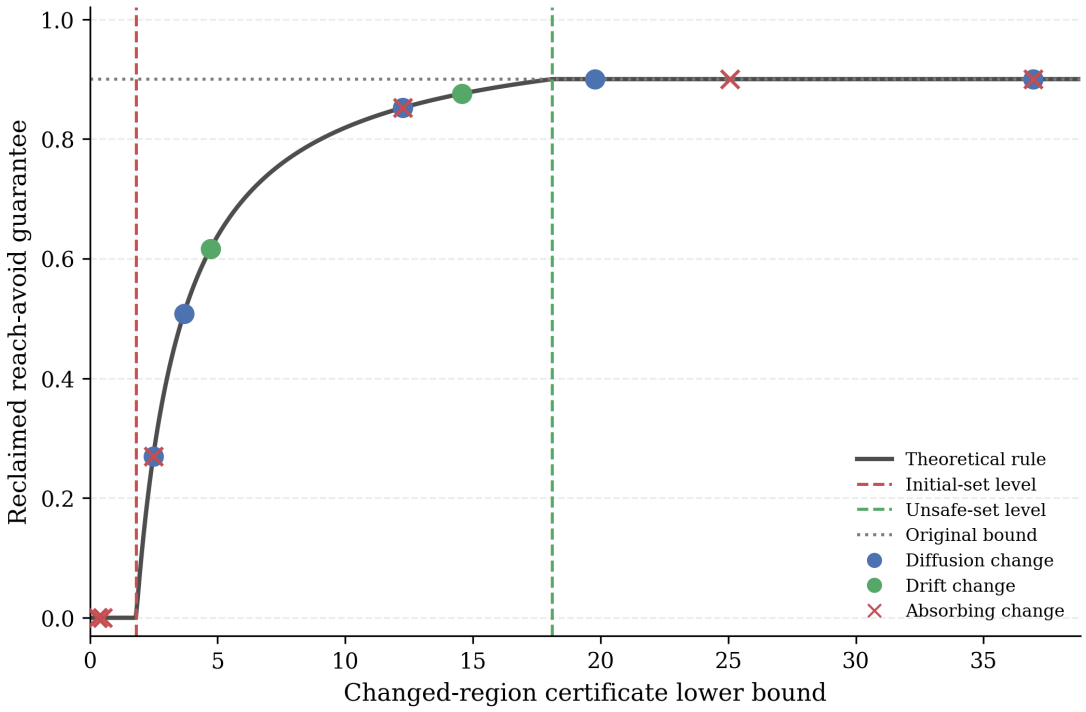

Figure 7: Continuous-time VeRecycle reclaiming rule as a function of the certified changed-region lower bound m_{lb}. The solid curve shows the theoretical rule $\rho_{\mathrm{rec}} = \min(1 - \alpha/\beta, \max(0, 1 - \alpha/m_{\mathrm{lb}}))$, and the markers show the evaluated scenarios. Below the initial-set level α, the reclaimed guarantee is zero; between α and β, it increases monotonically; and above the unsafe-set level β, it saturates at the original guarantee.

This behaviour has a simple state-space interpretation. Changes in areas where the certificate is small are safety-critical for the existing proof, so little or no guarantee can be reclaimed. Changes within regions where the certificate is large preserve

most or all of the original margin. Thus, the empirical results are consistent with the theoretical characterisation from RQ1: the reclaimed guarantee is governed by the local certificate minimum over X_q.

As a sanity check, we also compare the IBP lower bound with a grid-based approximation of the certificate minimum. The grid-based estimate uniformly discretises X_q and evaluates V at the grid points. This approximation is feasible here because the pendulum state space is two-dimensional, but it is not used in the certified pipeline. In higher dimensions, dense grid evaluation would be replaced by sound relaxation-based estimators such as IBP, stronger bounded-verification methods, or adaptive partitioning schemes. The corresponding comparison is reported in Appendix A.1.

5.3 Scenario-based comparison

This subsection further illustrates the theoretical characterisation from RQ1 and addresses the re-certification comparison in RQ2 by comparing VeRecycle with re-certification across representative non-absorbing perturbation regimes. The spatial locations of these scenarios are shown in Figure 6, where each region is labelled according to its role in the evaluation.

Figure 8 compares four representative scenarios. These are the same scenarios highlighted in Figure 6. The low-margin and borderline margin cases modify diffusion in regions where the existing certificate has low or intermediate values. The strong drift corridor case modifies the drift in a region farther along the closed-loop flow. The near-original-bound case modifies diffusion in a high-certificate region. These cases illustrate the main behaviours established by the theory: VeRecycle returns a modest guarantee in low-margin regions, a larger partial guarantee in intermediate regions, and the original guarantee in high-margin regions.

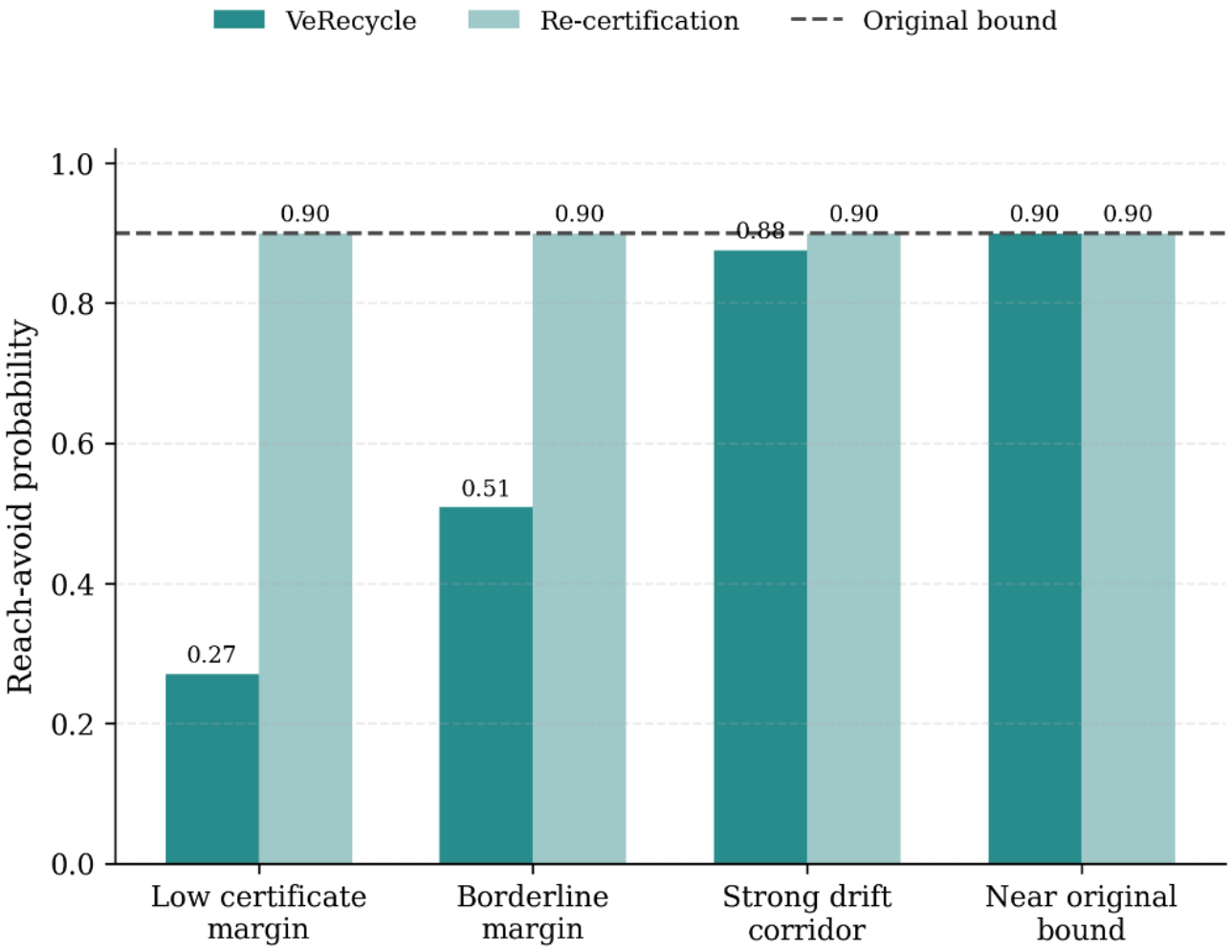

Figure 8: Representative non-absorbing scenarios. VeRecycle reuses the original certificate and returns an immediate certified bound from the local certificate lower bound, whereas re-certification retrains and verifies a new certificate for the modified system.

Table 2: Representative non-absorbing scenarios from the final evaluation. The m_{lb} values are reported in the raw checkpoint scale, where $\alpha = 1.8103$ and $\beta = 18.1032$. The re-certification column reports the fixed target reach–avoid probability requested from the retraining-and-verification pipeline, not an estimate of the best achievable certified bound for that scenario.

Scenario	m_{lb}	VeRecycle	Verified re-cert.	Re-cert. time (s)
Low certificate margin	2.480	0.270	0.900	69.99
Borderline margin	3.681	0.508	0.900	60.36
Strong drift corridor	14.568	0.876	0.900	55.41
Near original bound	36.949	0.900	0.900	48.92

This comparison illustrates the main practical distinction between the two methods. VeRecycle extracts all relevant information locally from the existing certificate, eliminating the need for global retraining. Re-certification, by contrast, optimises a new certificate for the modified dynamics. It can therefore certify the target reach-avoid probability when VeRecycle can only recover a smaller bound, but only after a substantially more expensive training and verification loop.

Table 2 reports the same representative non-absorbing scenarios quantitatively. In these runs, re-certification successfully verifies the target reach-avoid probability of 0.9 across all four scenarios. VeRecycle yields lower reclaimed guarantees in low- and intermediate-margin regions because it keeps the original certificate fixed and can only reclaim the guarantee supported by the local lower bound m_{lb}. In the near-original-bound case, VeRecycle already recovers the entire original target guarantee, so the two workflows report the same certified probability.

For re-certification, the reported value is the reach-avoid probability that the retraining-and-verification pipeline is asked to certify. In our runs, this target is 0.9. If re-certification returns a valid certificate, we report $\rho_{\mathrm{recert}} = 0.9$; if it fails, times out, or is unsupported, we report $\rho_{\mathrm{recert}} = 0$. We do not use sampled certificate-level ratios as certified guarantees.

Taken together, these results are consistent with the local-minimum reclaiming rule across qualitatively different non-absorbing regimes. They also illustrate the trade-off relevant to RQ2: VeRecycle provides an immediately reusable certified guarantee, while re-certification can recover the target certified probability at the cost of retraining and full verification. The conservatism of VeRecycle in this comparison is therefore structural: it derives from reusing a fixed certificate under arbitrary local changes, rather than from a numerical artefact of a particular experiment.

Absorbing sanity check. As an additional sanity check, we also considered an absorbing modification in a changed region with $m_{\mathrm{lb}} \leq \alpha$. This corresponds to the zero-recovery regime of the reclaiming rule, and continuous-time VeRecycle indeed returns the trivial bound $\rho_{\mathrm{rec}} = 0$.

5.4 Runtime analysis

This experiment addresses RQ2 by evaluating whether continuous-time VeRecycle provides a computational benefit over full re-certification when the fixed neural certificate is reused.

In this benchmark, the computational advantage of VeRecycle is substantial. Whereas re-certification requires iterative optimisation together with repeated verification steps, VeRecycle evaluates the existing certificate locally using IBP. In our experiments, VeRecycle consistently runs in sub-second time, whereas successful re-certification requires tens of seconds. This corresponds to orders-of-magnitude speedups.

The differences between the re-certification runtimes across scenarios should be interpreted as differences in optimisation and verification effort, not as a direct function of the reclaimed VeRecycle guarantee. Re-certification retrains a new certificate and then verifies global certificate conditions, so its runtime depends on how quickly training finds a candidate that can be verified. The two middle scenarios have similar runtimes because both require a comparable number of training and verification iterations before a valid certificate is found. The near-original-bound case is faster in this run, while the low-margin case takes longest. By contrast, VeRecycle does not retrain and only evaluates the fixed certificate on the changed region, which explains why its runtime remains much smaller and comparatively stable.

As described in Section 5.1, all runtime measurements use the same CPU-based Python implementation and execution environment. The absolute wall-clock times are therefore hardware-dependent, but the comparison is controlled because VeRecycle and re-certification are measured in the same software and execution environment.

The scaling behaviour of the two workflows is different. Re-certification scales with the cost of retraining a certificate and repeatedly verifying global certificate conditions over the full state space. This cost can increase with the system's size, the size of the neural networks, the complexity of the dynamics, and the verification partition. VeRecycle avoids retraining and verifies only the existing certificate over the changed region. Its cost, therefore, scales primarily with the size and partition complexity of X_q, as well as with the cost of propagating IBP bounds through the fixed certificate network. This does not remove the usual dimensional challenges of neural-network verification, but it localises the expensive part of the computation to the changed set.

VeRecycle is therefore particularly suitable when safety guarantees must be updated quickly after a localised model change, for example, in iterative controller design or adaptive systems where local dynamics may change. Re-certification remains useful when recovering the target certified probability is more important than the update time.

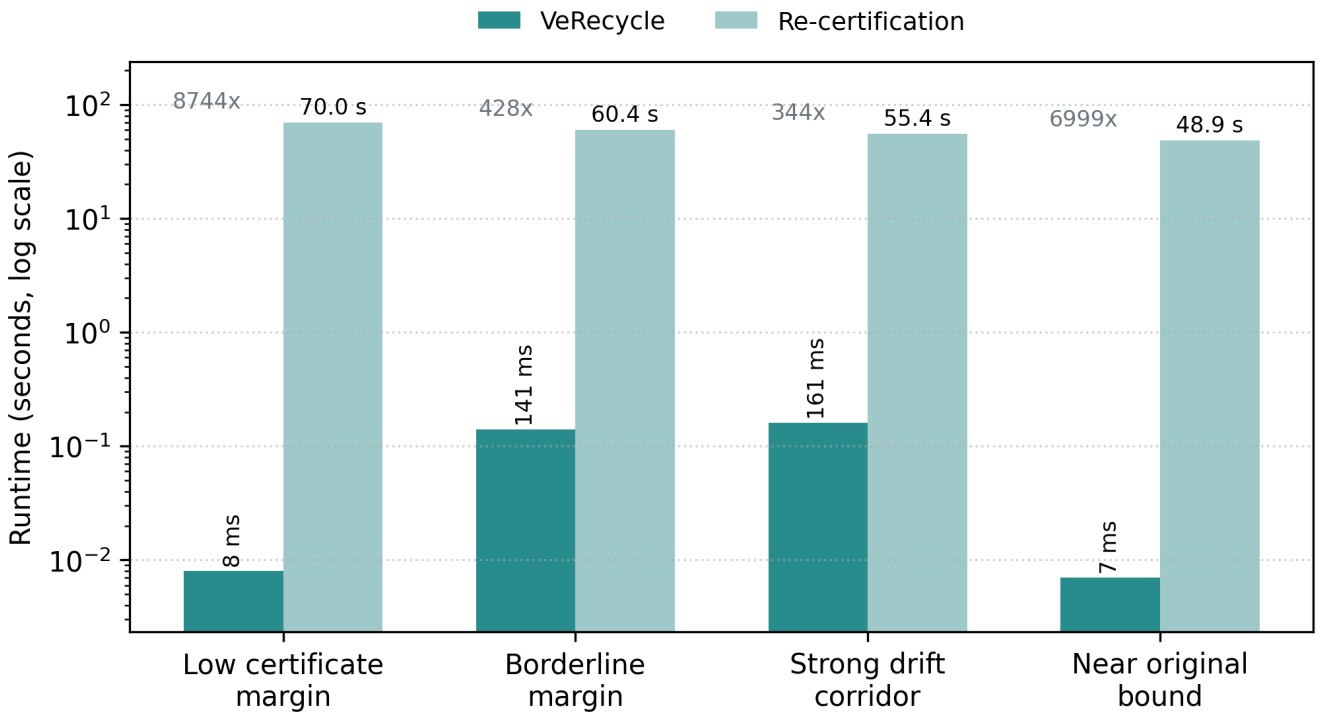

Figure 9: Runtime comparison between VeRecycle and re-certification. VeRecycle only computes a local IBP lower bound on the fixed certificate, whereas re-certification retrains and verifies a new certificate. Differences between re-certification runtimes reflect optimisation and verification effort for each modified scenario.

These runtime results deal with the computational part of RQ2: continuous-time VeRecycle can be implemented efficiently in practice and enables rapid safety reassessment after local model changes.

5.5 Continuous-time vs naive discretise-then-reclaim

This experiment addresses the discretise-then-reclaim part of RQ2. The question is not whether discrete-time VeRecycle is correct for a genuine discrete-time model, but whether a naive discretisation of a continuous-time system preserves the changed region X_q well enough for a discrete-time reclaiming rule to be applied safely. Thus, the comparison is not against the discrete-time VeRecycle theorem itself, but against an approximation workflow that first replaces the continuous-time problem by a coarse sampled representation.

We compare continuous-time VeRecycle with a deliberately naive discretise-then-reclaim approximation. This approximation should not be interpreted as a sound abstraction; rather, it represents the common shortcut of sampling the state space and applying the reclaiming rule to the sampled representation. The approximation replaces the continuous state space with a grid of width h and represents X_q only by the grid-centre points that fall inside it:

$$X_q^h = G_h \cap X_q.$$

It then evaluates

$$m_h = \min_{x \in X_q^h} V(x)$$

whenever X_q^h is nonempty. If $X_q^h = \emptyset$, this approximation treats the changed region as absent and returns the original guarantee.

This shortcut can fail because of spatial aliasing. A thin, changed region may exist in the continuous state space and may be encountered by trajectories, even though it contains no grid centres at a coarse resolution. In that case, the discretised representation removes the changed region before the reclaiming rule is applied.

The certificate-level consequence is summarised quantitatively in Table 3, after the main visual comparison in Figure 11. The thin corridor lies in a low-certificate region, with $m_{\mathrm{lb}} < \alpha$, so continuous-time VeRecycle returns $\rho_{\mathrm{rec}}^{\mathrm{CT}} = 0$. Additional certificate-level context is shown in Appendix A.3. By contrast, when the coarse grid misses the changed region, the naive baseline treats the change as absent and returns the original guarantee.

These results do not imply that discrete-time VeRecycle is unsound for genuine discrete-time models. If the system, certificate, transition relation, and changed set are all specified at the discrete-time level, then the discrete-time reclaiming rule

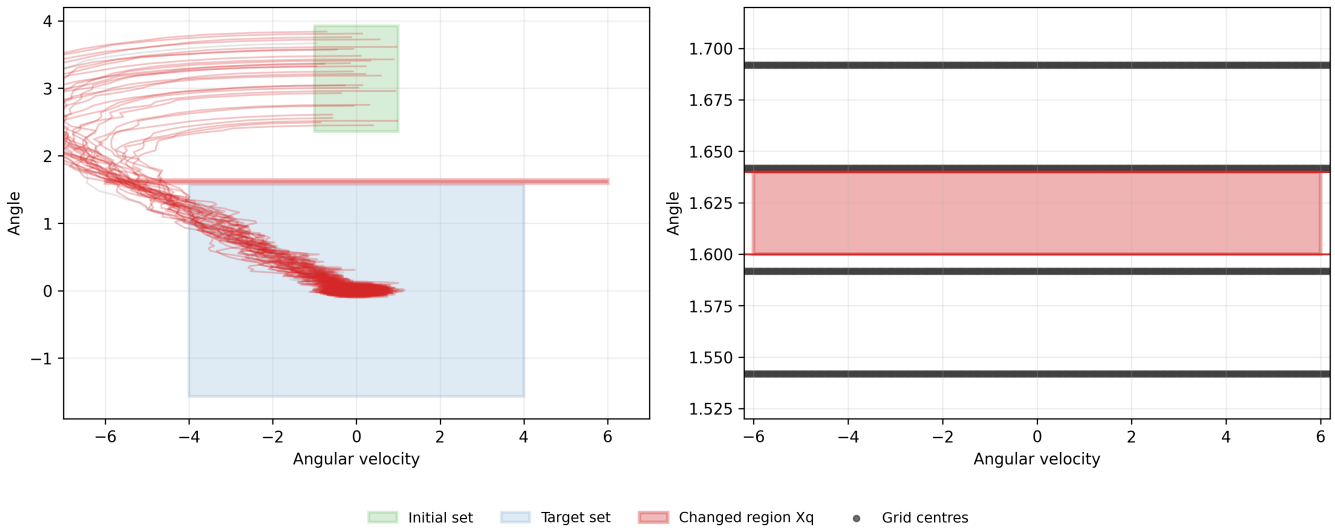

Figure 10: Spatial aliasing in a naive discretise-then-reclaim baseline. The thin changed region intersects closed-loop trajectories, but for a coarse grid, no grid centres lie inside the band, so the discrete representation treats the change as absent.

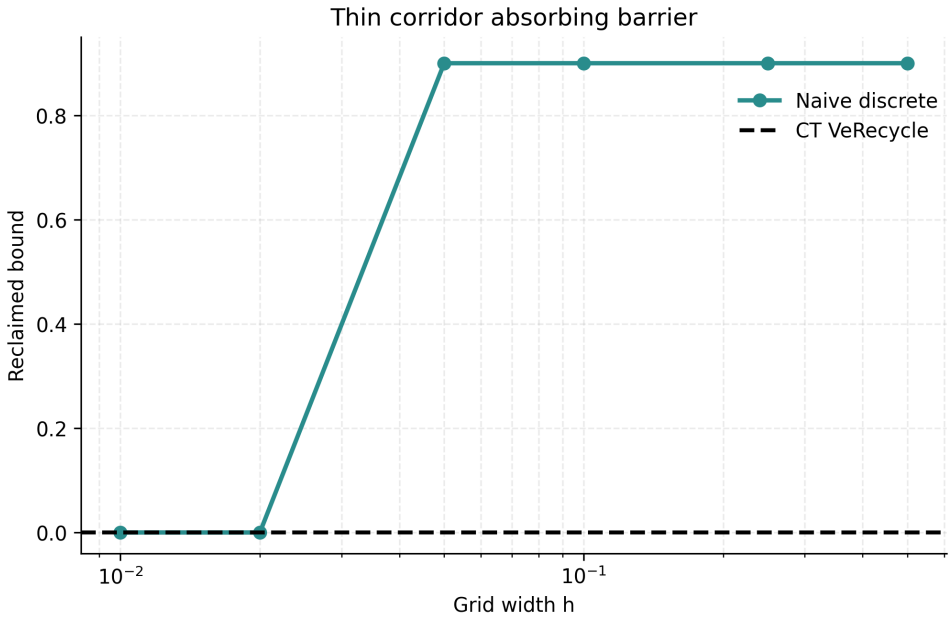

Figure 11: Continuous-time VeRecycle versus the naive discretise-then-reclaim baseline. The naive bound jumps when the grid-centre representation misses X_q, whereas the continuous-time bound stays sensitive to the changed region.

Table 3: Naive discretise-then-reclaim approximation for the thin corridor absorbing barrier. The failure occurs when the grid-centre representation of the changed region becomes empty.

Grid width h	$\lvert X_q^h \rvert$	$\rho_{\mathrm{rec}}^{\mathrm{CT}}$	ρ_{naive}
0.01	4800	0.0	0.0
0.02	1200	0.0	0.0
0.05	0	0.0	0.9
0.10	0	0.0	0.9
0.25	0	0.0	0.9
0.50	0	0.0	0.9

applies to that discrete model. The failure shown here concerns a different workflow: first approximating a continuous-time changed region by a coarse sampled representation and then applying a discrete-time reclaiming rule to that approximation. This shortcut is appropriate only when the discretisation soundly preserves the changed region and all relevant trajectory interactions required for the guarantee. If the discretisation removes X_q spatially or misses crossings temporally, then the reclaimed guarantee is computed for the wrong model. Continuous-time VeRecycle avoids this failure mode by operating directly on the continuous changed set.

Evaluation limitations. The main limitation of VeRecycle is that it can only reuse information already present in the fixed certificate. If the original certificate assigns low values to a changed region, VeRecycle may return a small or zero guarantee even when a different, newly trained certificate could certify a stronger property for the modified system. This is not a numerical failure, but an effect of reusing the original proof object. The method is therefore most useful when the changed region lies in a part of the state space where the existing certificate has sufficient margin. A second practical limitation is that computing m_{lb} requires a sound lower-bound estimator for the certificate over X_q. In this work, we use IBP because the benchmark is low-dimensional and the certificate network is compatible with the existing verification pipeline. For higher-dimensional systems, tighter relaxations, adaptive partitioning, or problem-specific set representations may be needed.

In practice, VeRecycle is best suited for rapid reassessment after local model changes when a certificate already exists, and the changed region is expected to retain certificate margin. Recertification is preferable when the changed region lies in a low-certificate area, when the recovered bound is too conservative for the application, or when a stronger guarantee is worth the additional training and verification cost.

Summary. Overall, the evaluation shows that the implementation behaves consistently with the derived reclaiming rule and is practically efficient on the benchmark considered. The reclaimed guarantees follow the theoretical dependence on the local certificate minimum established in RQ1. They also address RQ2 by showing computational savings over full re-certification and by exposing over-reclaiming caused by naive discretisation of local continuous-time changes. These results suggest that certificate reuse can be a useful component in iterative controller design and adaptive verification workflows, where local model changes occur more frequently than full safety proofs can be reconstructed from scratch.

6 Discussion

This chapter discusses what the thesis contributes relative to prior work, how the theoretical and experimental results should be interpreted, and where the limits of the present framework lie. The background chapter introduced the main ingredients separately. Here, the focus shifts from explaining those components to understanding what happens once they are brought together in the continuous-time VeRecycle setting.

6.1 Relation to prior work

Relation to VeRecycle. The most direct point of comparison is VeRecycle (Lutz et al., 2025), which introduced the reclaiming principle for discrete-time stochastic dynamical systems. The central insight of that work is that after a localised model change, part of the original certificate-based guarantee may still be recovered without retraining the certificate, and that the recovered guarantee is governed by the certificate values on the changed region.

The present thesis preserves that intuition, but transfers it to a continuous-time stochastic setting. This is not a purely notational reformulation. In VeRecycle, the reasoning is expressed through discrete stochastic transitions and one-step certificate conditions. In contrast, the present work reasons through stochastic differential equations and infinitesimal-generator conditions. The contribution of this thesis is therefore to show that the reclaiming idea is not an artefact of discrete-time modelling, but survives under the proof rule used for continuous-time stochastic certificates.

Relation to continuous-time neural certificates. The second main line of prior work is the continuous-time neural certificate framework of Neustroev et al. (2025). That work provides the continuous-time probabilistic proof rule on which this thesis relies. In particular, it shows how reach-avoid guarantees can be certified for stochastic differential equations using generator-based supermartingale conditions.

The present thesis does not replace that framework, but builds directly on it. The question addressed here is different. Neustroev et al. studies how to certify a fixed continuous-time stochastic system. This thesis studies what can still be certified after the system changes locally, while the certificate is kept fixed. In that sense, the present work complements continuous-time neural certification with a reuse mechanism for already-certified systems.

Relation to the broader certificate literature. More broadly, this thesis sits within the line of work that uses Lyapunov, barrier, and supermartingale-style certificates to verify nonlinear and stochastic systems (Kalman and Bertram, 1960; Prajna et al., 2004, 2007; Neustroev et al., 2025). What distinguishes the present contribution from that wider literature is not the introduction of a new certificate class, but the characterisation of how an existing certificate can still support guarantees after a localised change in the dynamics.

6.2 What continuous-time VeRecycle adds

The main contribution of this thesis is that it identifies a clean continuous-time reclaiming rule that depends on the changed region only through the quantity

$$m := \inf_{x \in X_q} V(x).$$

Once the original certificate V is fixed, the maximal reclaimed guarantee certifiable by reusing this fixed certificate is governed by one local geometric feature of that certificate: its smallest value on the region where the dynamics may have changed.

This is useful both mathematically and practically. Mathematically, it shows that the continuous-time reclaiming problem admits a closed-form characterisation rather than only a qualitative statement. Practically, it means that one does not need to retrain the certificate or reanalyse the whole state space from scratch in order to obtain a sound post-change bound.

The inverted-pendulum benchmark helps illustrate why this is meaningful. The certificate acts as a risk landscape over the pendulum state space, whose coordinates are angular velocity and angle. A localised change in the drift or diffusion does not automatically invalidate all information encoded by that landscape. What matters is whether the changed region already lies in a part of the state space that the certificate regarded as relatively risky or relatively safe. The quantity m captures exactly that distinction.

6.3 Interpretation of the theoretical result

The main theorem shows that the reclaimed threshold has three qualitatively different regimes.

Low-infimum regime. When $m \leq \alpha$, the reclaimed bound becomes trivial. Intuitively, this means that the changed region reaches certificate levels that are as low as the initial certified region. In that case, the proof cannot rule out that the modified dynamics exploit precisely the part of the state space that the original certificate treated as comparatively safe.

Intermediate regime. When $\alpha < m < \beta$, the reclaimed bound is strictly positive but strictly below the original bound. This is the genuinely interesting regime. Here, the changed region is not so benign that the original guarantee survives intact, but it is also not so risky that all nontrivial reclaiming becomes impossible. In this regime, the theorem quantifies exactly how much of the original certified guarantee can still be certified from the fixed certificate.

Recovery regime. When $m \geq \beta$, the original certified bound is fully recovered by the fixed-certificate proof rule. In this case, the fixed-certificate proof rule recovers the original certified bound. It means that every state in the changed region already lies beyond the unsafe threshold of the original certificate, so the certificate theorem never needs to rely on the generator condition there. From the perspective of the proof, the local change then becomes irrelevant to the certified threshold.

This three-regime picture gives a simple interpretation of the result. It makes the reclaiming rule easy to interpret and also easy to communicate: the modified region is either too low to support reclaiming, high enough to support partial reclaiming, or high enough to preserve the original guarantee completely.

6.4 Interpretation of the experimental results

The experiments in Chapter 5 serve two purposes. First, they test whether the empirical implementation behaves consistently with the theoretical reclaiming rule. Second, they assess whether the resulting guarantees are useful in practice.

The results indicate that the implementation follows the theoretical dependence on m closely. In particular, the recovered guarantees vary in the way predicted by the theorem: lower values of m lead to weaker recovered guarantees, while larger values of m lead to stronger recovery, up to full recovery once the certificate infimum exceeds the relevant threshold. This agreement is important, because it shows that the theorem is not merely a formal statement detached from the implemented method.

The inverted-pendulum benchmark makes this dependence on m visible in a concrete state space. Moving X_q to different angle–velocity regions changes the local certificate minimum, and the reclaimed guarantee changes accordingly. Regions with low certificate values lead to weak or trivial recovery, while regions with high certificate values allow most or all of the original guarantee to be reclaimed. This illustrates the theorem's mechanism without treating the benchmark as evidence for a new theoretical claim.

A second important outcome is computational. The point of reclaiming is not only that it is possible in principle, but that it can be substantially cheaper than recomputing a certificate from scratch. The evaluation therefore supports the practical motivation of the thesis: when the change is localised, reuse can provide a meaningful and efficient alternative to full re-certification.

6.5 Limitations

The best perspective is to reflect on the limitations of this thesis relative to its two main contributions. The first contribution is theoretical: it characterises the maximal reach–avoid guarantee that can be certified by reusing a fixed continuous-time certificate after a localised model change. The second contribution is empirical: it evaluates this reuse mechanism on a stochastic inverted-pendulum benchmark and compares it with re-certification and a naive discretise-then-reclaim workflow.

Limitations of the theoretical contribution. The theoretical result answers RQ1 within the formal setting defined in Chapter 3. In that setting, the original system is a time-homogeneous continuous-time stochastic system, the fixed certificate V satisfies the reach–avoid certificate conditions, and the modified dynamics agree with the original dynamics outside the changed region X_q. Under these assumptions, Chapter 4 proves both directions of the reclaiming rule: the proposed bound is certifiable by reusing V, and no larger bound can be certified uniformly from the same certificate for all admissible local modifications supported on X_q.

This is a maximality result for *fixed-certificate reuse*. It should not be interpreted as a statement about the true reach–avoid probability of the modified system. A particular modified system may, in reality, satisfy a stronger property than the reclaimed bound suggests. It may also be possible to certify a stronger guarantee by training a new certificate for the modified dynamics. The theorem deliberately does not address that question: it characterises what can still be justified without changing the original proof object.

The result is also conservative because it is uniform over all admissible modifications inside X_q. This is useful when the local change is only partially known, but it means that the reclaiming rule does not exploit detailed information about the modified drift or diffusion inside the changed region. If the exact modified dynamics are known, a less conservative model-specific reuse rule may be possible. Such a rule would require additional assumptions and is outside the scope of this thesis.

Finally, the theoretical development is restricted to the reach–avoid fragment of the continuous-time certificate framework. The full framework of Neustroev et al. (2025) also supports reach-stay-avoid specifications. Extending the same reclaiming idea to the stay component, to time-varying target and unsafe sets, or to more general certificate conditions would require additional work. The present theorem therefore establishes the continuous-time VeRecycle principle for the reach–avoid setting considered here, but not for every possible continuous-time certificate specification.

Limitations of the empirical contribution. The empirical contribution answers RQ2 only at the level of the benchmark and implementation studied in Chapter 5. The stochastic inverted pendulum is useful because it gives a concrete continuous-time system with a neural policy, a neural certificate, and visually interpretable changed regions. However, it is still a two-dimensional benchmark. The experiments therefore show that continuous-time VeRecycle is feasible in this setting, but they do not establish scalability to high-dimensional systems, larger neural networks, or more complex state-space geometries.

A second limitation is that the changed regions are selected to probe the regimes predicted by the theory: no recovery, partial recovery, and full recovery. This is appropriate for evaluating whether the implementation follows the theoretical rule, but it is

not an exhaustive study of all physically plausible pendulum perturbations. The experiments vary both the location of X_q and the type of local dynamics modification, but only within the finite set of scenarios reported in the evaluation.

The computation of m_{lb} is another practical limitation. The theorem depends on the true infimum

$$m = \inf_{x \in X_q} V(x),$$

whereas the implementation uses a sound IBP lower bound m_{lb}. This preserves soundness, but it can make the recovered guarantee more conservative. In the two-dimensional pendulum benchmark, local mesh refinement is still manageable. Within higher-dimensional systems, the number of boxes may grow quickly, and tighter relaxations or adaptive partitioning strategies may be needed to obtain useful bounds efficiently.

The comparison with re-certification should also be interpreted carefully. VeRecycle and re-certification answer different practical questions. VeRecycle asks what can be certified immediately by reusing the original certificate, whereas re-certification trains and verifies a new certificate for the modified system. In the experiments, re-certification is evaluated as a success-or-failure attempt to certify a fixed target probability. The comparison therefore demonstrates the practical trade-off between fast reuse and full re-certification in this implementation, but it does not prove that VeRecycle will always outperform every possible re-certification pipeline.

Finally, the naive discretise-then-reclaim experiment is intentionally limited. Its purpose is to illustrate a failure mode: a coarse sampled representation can miss a continuous changed region and then over-reclaim. This does not imply that discrete-time VeRecycle is unsound, nor that all discretisation-based approaches are inadequate. It shows only that applying a discrete-time reclaiming rule to a continuous-time system requires a sound abstraction of the changed region and of the relevant continuous-time interactions.

Overall, the result is complete within the fixed-certificate reach–avoid setting considered in this thesis, but its scope is deliberately limited. The theorem applies to localised changes in continuous-time models and is evaluated on a single benchmark. Future work could extend the reclaiming rule to reach–stay–avoid specifications, incorporate partial knowledge of the modified dynamics inside X_q, and develop scalable lower-bound computation methods for higher-dimensional systems.

6.6 Chapter summary

This chapter positioned the thesis relative to prior work and interpreted the main findings. Relative to Lutz et al. (2025), the present work expands reclaiming from discrete-time stochastic systems to continuous-time stochastic dynamics. Relative to Neustroev et al. (2025), it adds a principled certificate-reuse mechanism after localised model change. The main result is especially interesting because it delivers a simple and interpretable reclaiming rule governed by the local certificate infimum m. At the same time, the framework remains conservative, fixed-certificate, and proof-rule dependent. These limitations matter for responsible use, and they motivate the broader-impact and reflection chapter that follows.

7 Broader impact and reflection

This chapter reflects on the wider significance of the thesis for computer science and, more specifically, for trustworthy autonomous and learning-enabled systems. The main impact considered here is the extension of VeRecycle from discrete-time stochastic systems to continuous-time stochastic dynamics. This matters because many safety-critical autonomous systems evolve continuously in time, while existing certificate-reuse methods were formulated for discrete-time models. The chapter also discusses the limitations of the proposed framework from a responsible-use perspective and reflects on the research process itself.

7.1 Broader impact in computer science and safe autonomy

A main challenge in computer science nowadays is how to combine increasingly capable learning-based systems with guarantees strong enough for safety-critical use. In many applications, such as robotics, industrial automation, or autonomous navigation, it is not enough for a controller to perform well on average. One must also reason about rare but unacceptable failures.

This thesis contributes to that broader challenge by studying how formal guarantees can remain useful after a system changes, specifically in the continuous-time setting. The particular impact of the work is that it extends the VeRecycle reclaiming principle, originally formulated for discrete-time stochastic systems, to stochastic differential equations. This means that certificate reuse is no longer tied only to sampled transition models, but can also be studied for systems whose dynamics are described by drift, diffusion, and infinitesimal-generator conditions. In practice, this gives a more realistic view of verification for physical autonomous systems: one that acknowledges both that systems evolve continuously and that their dynamics may change locally over time.

A discrete-time certificate-reuse result can be useful for sampled models, but it does not by itself explain what can be reclaimed for a stochastic process evolving between sampling instants. By deriving the reclaiming rule directly for continuous-time certificates, this thesis addresses that distinction explicitly: the reused guarantee is justified through infinitesimal-generator conditions rather than one-step transition conditions.

7.2 Practical relevance of continuous-time reclaiming

The practical relevance is especially clear in the stochastic inverted-pendulum evaluation used in this thesis. If the drift or diffusion changes only in a local part of the pendulum state space, it may be unnecessary and expensive to retrain a new certificate immediately. At the same time, it is also unsatisfactory to pretend that the original guarantee still holds unchanged. Continuous-time VeRecycle provides a middle ground: it reuses the fixed certificate and computes the guarantee that can still be justified from the certificate values on the changed region.

The reclaiming viewpoint is useful because it separates what can still be certified from what would require a new certificate. In safety-critical settings, having a conservative but explicit reclaimed guarantee may be far more useful than having no guarantee at all, or an invalid guarantee that is mistakenly trusted.

This perspective can also be useful beyond the specific reach–avoid setting studied here. Instead of treating a model change as a reason to discard an existing certificate immediately, the framework asks how much of the original guarantee can still be justified. This idea could support future work on adaptive verification, runtime verification, and certificate reuse for learned systems.

7.3 Limitations and responsible use

At the same time, the framework should not be used more broadly than the theorem justifies.

First, reclaiming is not the same as full re-certification. The method does not compute the exact reach-avoid probability of the modified system. Instead, it computes the maximal guarantee that can still be certified from the original certificate under the stated assumptions. This distinction matters. A reclaimed bound is a sound lower bound, not a complete description of post-change behaviour.

Second, the result applies only to localised changes supported on a known region X_q, under the admissibility assumptions defined in the thesis. If the real change is not localised in that sense, or if the policy itself changes, or if the system violates the required regularity assumptions, then the framework no longer applies as stated.

Third, the method is deliberately conservative. This is useful because the guarantee holds for every admissible localised change, without needing to know the exact modified dynamics. However, it also means that the reclaimed bound may be lower than what is actually true for a specific concrete change. Responsible use therefore requires treating the result as a worst-case certified lower bound, not as an estimate of the true reach–avoid probability.

In this sense, the framework is best viewed as a principled fallback tool: not a replacement for full re-certification when that is feasible, but a rigorous way to preserve meaningful guarantees when immediate retraining is costly or unavailable.

7.4 Reflection on the research process

A major lesson from this thesis is that transferring an idea from discrete time to continuous time is rarely only a matter of rewriting notation. At first sight, the VeRecycle intuition appears conceptually similar in both settings: one reuses a certificate

and studies how changes to the region affect the guarantee. But the actual proof mechanisms are different. In discrete time, the reasoning is tied to one-step stochastic transitions. In continuous time, it depends on infinitesimal-generator conditions. Making that bridge precise required much more care than the high-level analogy initially suggests.

A second lesson concerns the role of notation and explanation in theoretical work. One of the recurring challenges in this thesis was not only to derive a correct result, but also to present it in a way that remains readable. This became especially clear when developing the background, the problem formulation, and the proof chapter. For formal work to be convincing, the mathematics has to be correct, but the reader must also be able to see the idea behind it. The running example, the explicit notation table, and the chapter-level reading guides were all attempts to make the argument easier to follow without weakening its correctness.

A third reflection is that the thesis reinforced the value of combining formal methods with learning-based verification rather than treating them as separate worlds. Neural certificates make certification scalable to systems where analytic certificates are hard to obtain. Formal proof rules make those learned objects trustworthy. The present work adds that such guarantees can also remain partially reusable after structured change. For me, that combination is one of the most interesting directions in the area: not learning instead of verification, and not verification despite learning, but learning and verification working together.

7.5 Chapter summary

The wider impact of this thesis lies in showing that formal guarantees for learning-enabled stochastic systems do not have to be treated as all-or-nothing objects once the model changes. The continuous-time reclaiming rule allows a more realistic and reusable view of certification under uncertainty. At the same time, the framework must be used carefully: it provides a conservative reclaimed bound under explicit assumptions, not a full replacement for re-certification. The research process also highlighted how much careful translation, notation, and explanation matter when moving ideas across modelling settings. The next and final chapter returns to the thesis question directly and closes the thesis with a concise summary and outlook.

8 Conclusion

This thesis studied whether the VeRecycle reclaiming principle can be extended from discrete-time stochastic systems to continuous-time stochastic dynamical systems. More specifically, it asked whether the maximal universally valid reclaimed reach-avoid guarantee can still be characterised directly from a fixed certificate and the changed region, without retraining the certificate itself.

The main result shows that such a continuous-time reclaiming rule can be obtained. Building on the continuous-time certificate framework of Neustroev et al. (2025) and the reclaiming viewpoint of Lutz et al. (2025), the thesis derived a continuous-time reclaiming rule for localised model change. The result shows that once a continuous-time reach-avoid certificate V is fixed, the maximal reclaimed guarantee is governed by the original certified threshold and by the local quantity

$$m := \inf_{x \in X_q} V(x),$$

that is, the smallest certificate value attained on the changed region. In this sense, the thesis shows that the key insight behind VeRecycle survives the transition from discrete-time stochastic transitions to continuous-time stochastic differential equations.

Beyond the theorem itself, the thesis also contributed a conceptual model for interpreting reclaiming in continuous time. The running example made this intuition concrete: when a robot or pendulum-like system encounters a localised change in its dynamics, not all information carried by the original certificate is lost. What matters is how the original certificate already ranked that region in terms of risk. The reclaiming rule formalises exactly how that local certificate geometry constrains the guarantee that can still be justified.

The experimental evaluation supported this picture. On the benchmark considered in this thesis, the implemented method behaved consistently with the theoretical dependence on m . The experiments also showed the practical value of reclaiming compared with full re-certification: when the change is localised, the fixed certificate can provide a certified post-change bound without retraining a new certificate from scratch.

At the same time, the framework has clear boundaries. It is fixed-certificate, conservative, and restricted to admissible localised changes. These limitations are not accidental, they are part of what makes the guarantee sound. The present work therefore should not be understood as a replacement for full re-certification in all cases, but as a rigorous reuse mechanism for the structured setting studied here.

8.1 Future work

Several directions for future work follow from this thesis.

A first direction is to move beyond fixed-certificate reuse and study whether reclaiming can be combined with local certificate repair. The present framework deliberately asks what can be preserved without retraining. A next step would be to investigate how much additional gain is possible when one allows limited post-change refinement.

A second direction is to broaden the class of specifications. This thesis focused on the time-homogeneous reach-avoid fragment of the continuous-time framework. Extending reclaiming ideas to richer continuous-time objectives, including time-varying or reach-stay-avoid settings, would be a natural development.

A third direction is to explore less conservative use of model information. The theorem is uniform over all admissible modifications supported on X_q. If more is known about the concrete modified dynamics, sharper reclaimed bounds may be possible.

Finally, a practical direction is to evaluate continuous-time reclaiming on a wider range of systems and certificate architectures. The theorem is general, but its empirical behaviour should also be studied across broader benchmarks and implementation choices.

8.2 Final remark

This thesis shows that a formal guarantee does not always have to be discarded after a local model change. Under the assumptions studied here, part of the original guarantee can be recovered from the fixed certificate, weakened in a sound way, and still used as a certified lower bound. This is the role of continuous-time VeRecycle, and it is the main contribution of this thesis.

References

A. Abate, D. Ahmed, M. Giacobbe, and A. Peruffo. Formal synthesis of Lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2021a. doi: 10.1109/LCSYS.2020.3005328.

A. Abate, M. Giacobbe, and D. Roy. Learning probabilistic termination proofs. In *Proceedings of the 33rd International Conference on Computer Aided Verification (CAV 2021)*, volume 12760 of *Lecture Notes in Computer Science*, pages 3–26. Springer, 2021b. doi: 10.1007/978-3-030-81688-9_1.

A. Abate, M. Giacobbe, and D. Roy. Stochastic Omega-Regular verification and control with Supermartingales. In *Proceedings of the 36th International Conference on Computer Aided Verification (CAV 2024)*, volume 14683 of *Lecture Notes in Computer Science*, pages 395–419. Springer, 2024. doi: 10.1007/978-3-031-65633-0_18.

A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. In *Proceedings of the 18th European Control Conference (ECC 2019)*, pages 3420–3431. IEEE, 2019. doi: 10.23919/ECC.2019.8796030.

M. Ansaripour, K. Chatterjee, T. A. Henzinger, M. Lechner, and D. Zikelic. Learning provably stabilizing neural controllers for discrete-time stochastic systems. In *Proceedings of the 21st International Symposium on Automated Technology for Verification and Analysis (ATVA 2023)*, volume 14215 of *Lecture Notes in Computer Science*, pages 357–379. Springer, 2023. doi: 10.1007/978-3-031-45329-8_17.

T. Badings, W. Koops, S. Junges, and N. Jansen. Learning-based verification of stochastic dynamical systems with neural network policies, 2024.

C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008. ISBN 9780262026499.

R. M. Blumenthal and R. K. Getoor. *Markov Processes and Potential Theory*, volume 29 of *Pure and Applied Mathematics*. Academic Press, New York, 1968.

Y.-C. Chang, N. Roohi, and S. Gao. Neural Lyapunov control. In *Advances in Neural Information Processing Systems*, volume 32, pages 3240–3249. Curran Associates, Inc., 2019.

K. Chatterjee, T. A. Henzinger, M. Lechner, and D. Zikelic. A learner-verifier framework for neural network controllers and certificates of stochastic systems. In *Proceedings of the 29th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2023)*, volume 13993 of *Lecture Notes in Computer Science*, pages 3–25. Springer, 2023. doi: 10.1007/978-3-031-30823-9_1.

E. M. Clarke, O. Grumberg, D. Kroening, D. A. Peled, and H. Veith. *Model Checking*. MIT Press, 2 edition, 2018. ISBN 9780262038836.

C. Dawson, S. Gao, and C. Fan. Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 39(3):1749–1767, 2023. doi: 10.1109/TRO.2022.3232542.

M. Giacobbe, D. Kroening, and J. Parsert. Neural termination analysis. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022)*, pages 633–645. ACM, 2022. doi: 10.1145/3540250.3549120.

R. E. Kalman and J. E. Bertram. Control system analysis and design via the "second method" of Lyapunov: I—Continuous-Time systems. *Journal of Basic Engineering*, 82(2):371–393, 1960. doi: 10.1115/1.3662604.

H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ, 3 edition, 2002. ISBN 9780130673893.

S. Lutz, M. T. J. Spaan, and A. Lukina. VeRecycle: Reclaiming guarantees from probabilistic certificates for stochastic dynamical systems after change. In J. Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)*, pages 457–465. International Joint Conferences on Artificial Intelligence Organization, August 2025. doi: 10.24963/ijcai.2025/52. Main Track.

F. B. Mathiesen, S. C. Calvert, and L. Laurenti. Safety certification for stochastic systems via neural barrier functions. *IEEE Control Systems Letters*, 7:973–978, 2023. doi: 10.1109/LCSYS.2022.3229865.

G. Neustroev, M. Giacobbe, and A. Lukina. Neural Continuous-Time Supermartingale certificates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27538–27546, 2025. doi: 10.1609/aaai.v39i26.34966.

B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, Berlin, Heidelberg, 5 edition, 1998. doi: 10.1007/978-3-662-03620-4.

S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Proceedings of the 7th International Workshop on Hybrid Systems: Computation and Control (HSCC 2004)*, volume 2993 of *Lecture Notes in Computer Science*, pages 477–492. Springer, 2004. doi: 10.1007/978-3-540-24743-2_32.

S. Prajna, A. Jadbabaie, and G. J. Pappas. Stochastic safety verification using barrier certificates. In *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC 2004)*, pages 929–934. IEEE, 2004. doi: 10.1109/CDC.2004.1428804.

S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, 2007. doi: 10.1109/TAC.2007.902736.

J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, NJ, 1991. ISBN 9780130408907.

P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, New York, 2009. doi: 10.1007/978-1-4419-0224-5.

D. Zikelic, M. Lechner, T. A. Henzinger, and K. Chatterjee. Learning control policies for stochastic systems with Reach-Avoid guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11926–11935, 2023. doi: 10.1609/aaai.v37i10.26407.

A Additional Evaluation Results

A.1 IBP versus grid-based minimum

As a sanity check for the tightness of the IBP lower bound, we compare m_{lb} against a grid-based approximation of the true certificate minimum over X_q. The grid-based estimate is obtained by uniformly discretising X_q and evaluating V at each grid point. This comparison is feasible in the two-dimensional pendulum setting and is used only for diagnostic purposes.

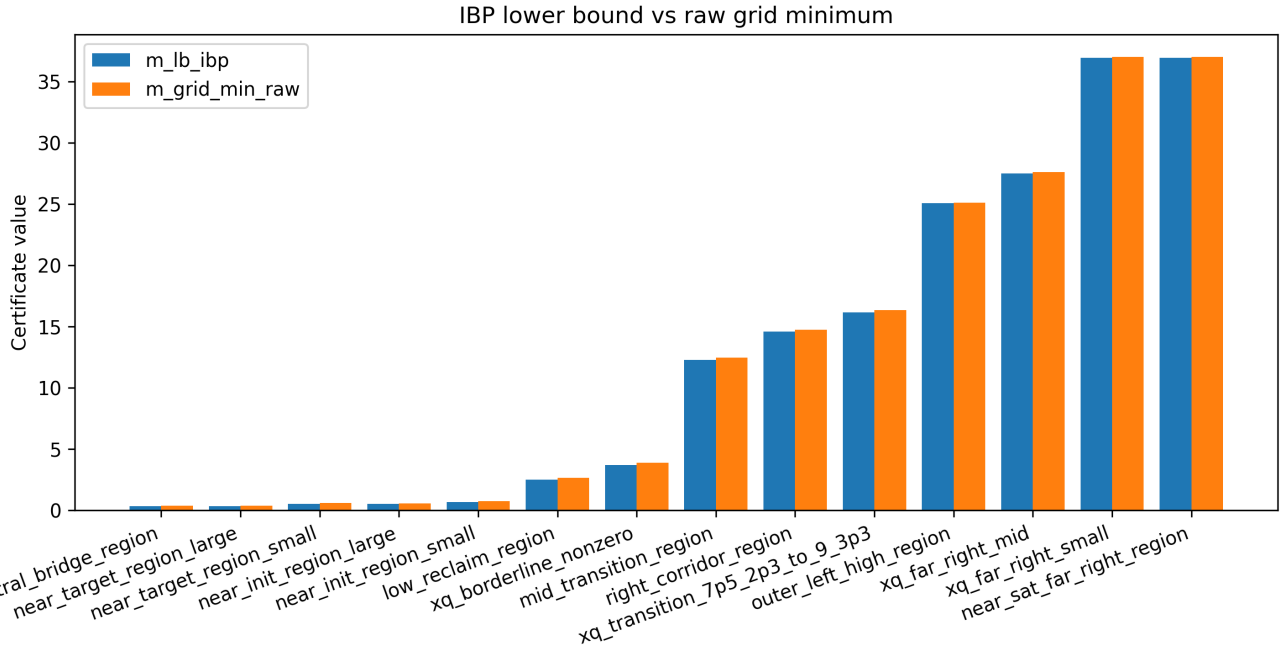

Figure 12: IBP lower bound versus grid-based estimate of $\inf_{x \in X_q} V(x)$. The close agreement indicates that the IBP bounds used in the main evaluation are sufficiently tight in this setting.

A.2 Temporal aliasing diagnostic

To illustrate the temporal component of discretisation error, we compare two empirical hit detectors on the same fine-resolution reference trajectories: a segment-based detector, which checks whether consecutive fine-resolution trajectory segments intersect X_q, and a sampled detector, which checks only states observed at a coarser sampling interval. This diagnostic is not part of the certified pipeline; it is included only to show how coarse temporal sampling can miss interactions with a thin changed region.

A.3 Certificate context for the thin-corridor aliasing example

Figure 14 gives the certificate-level context for the thin-corridor discretisation example from Section 5. The changed corridor lies in a low-certificate region, with $m_{\mathrm{lb}} < \alpha$. Continuous-time VeRecycle therefore returns the trivial reclaimed bound. The naive discretise-then-reclaim approximation can miss this region when no grid centres fall inside the corridor, causing the approximation to treat the local change as absent.

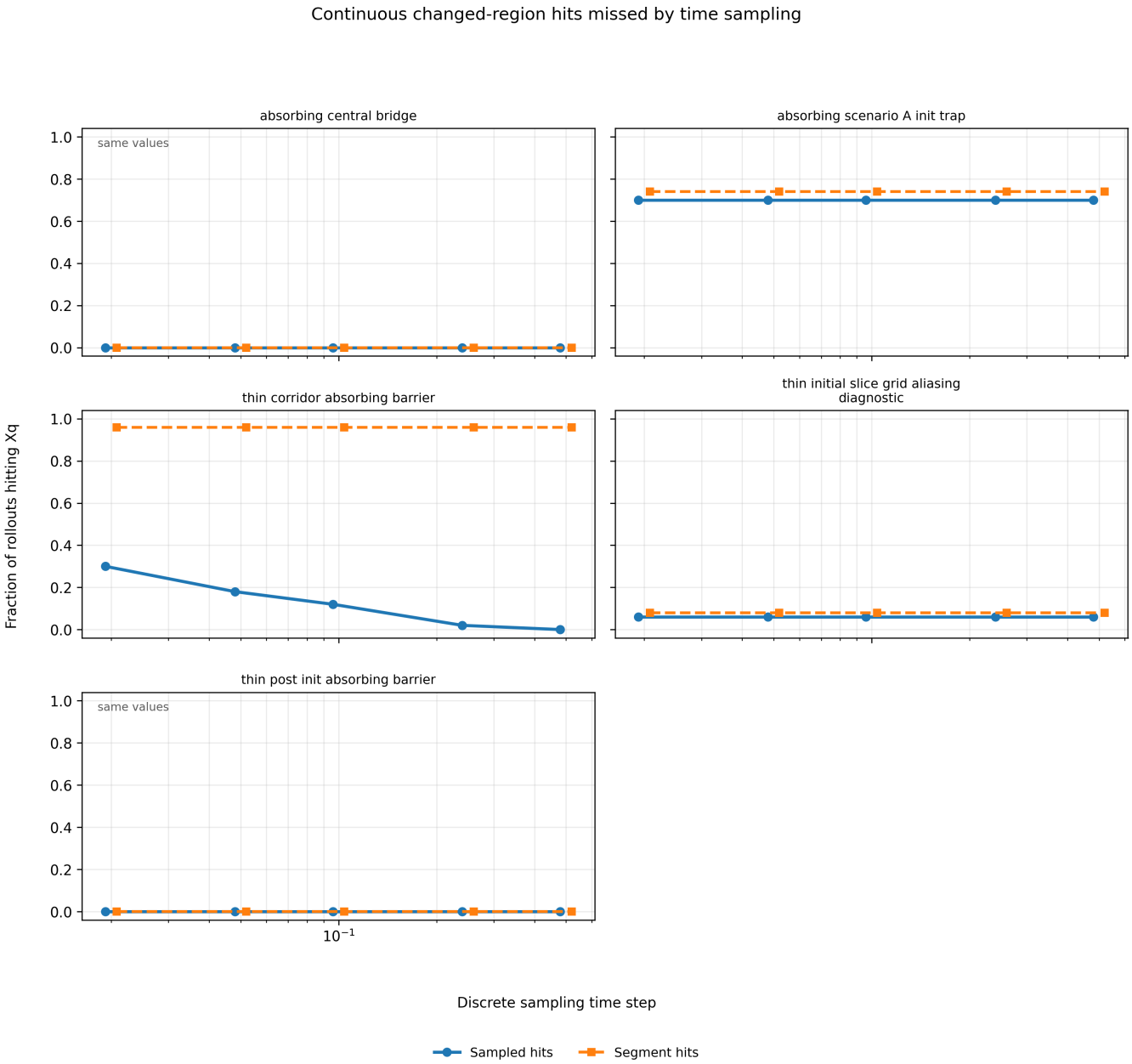

Figure 13: Empirical temporal-aliasing diagnostic. The segment-based detector checks whether fine-resolution trajectory segments intersect X_q, whereas the sampled detector checks only states observed at coarser sampling intervals.

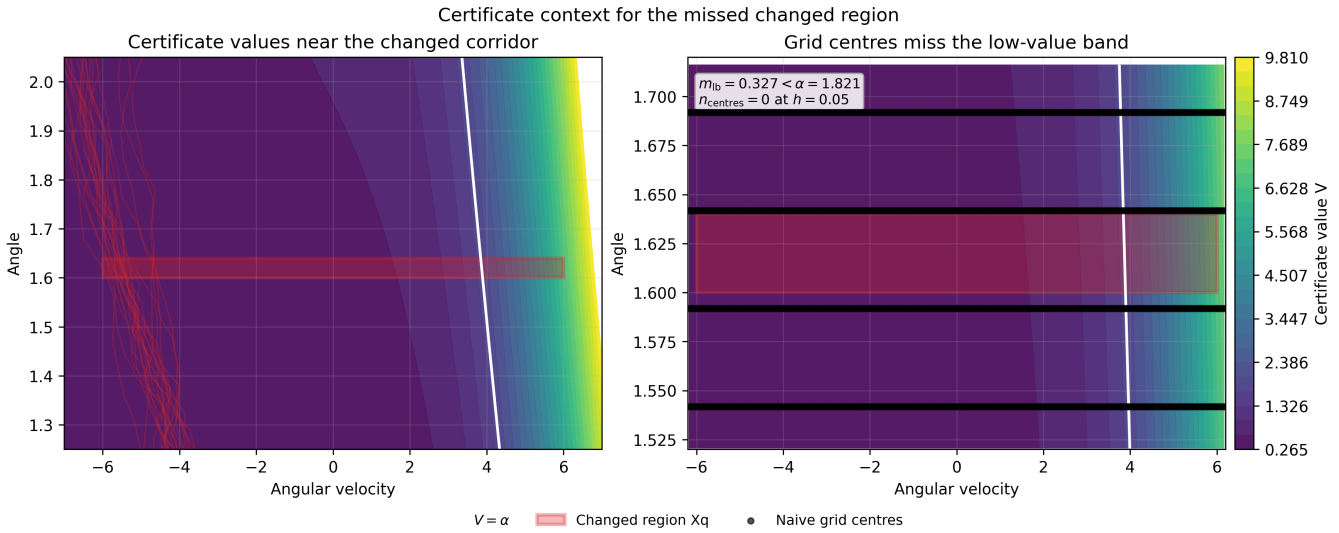

Figure 14: Certificate context for the thin-corridor aliasing example. The changed corridor lies in a low-certificate part of the landscape, while the coarse grid-centre representation may miss the changed region entirely.

Generative AI Use Disclosure

Generative AI tools were used to support parts of the research and writing process of this thesis. ChatGPT, using OpenAI language models available during the thesis process, was used for brainstorming, improving structure, rephrasing paragraphs, and drafting clearer academic prose. Grammarly was used for grammar, spelling, and style suggestions. Codex was used to support coding tasks, including generating and refining implementation ideas, debugging assistance, and improving code structure.

These tools were used as support tools only. They were not used as independent sources of scientific claims, mathematical results, or experimental conclusions. All theoretical developments, implementation choices, experiments, results, and interpretations were reviewed, checked, and verified by the author. The author remains fully responsible for the accuracy, originality, and integrity of the content of this thesis.