

Hochschule Karlsruhe

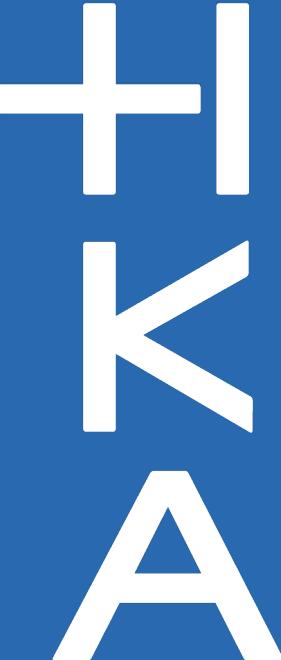
University of

Applied Sciences

Fakultät für

Maschinenbau und

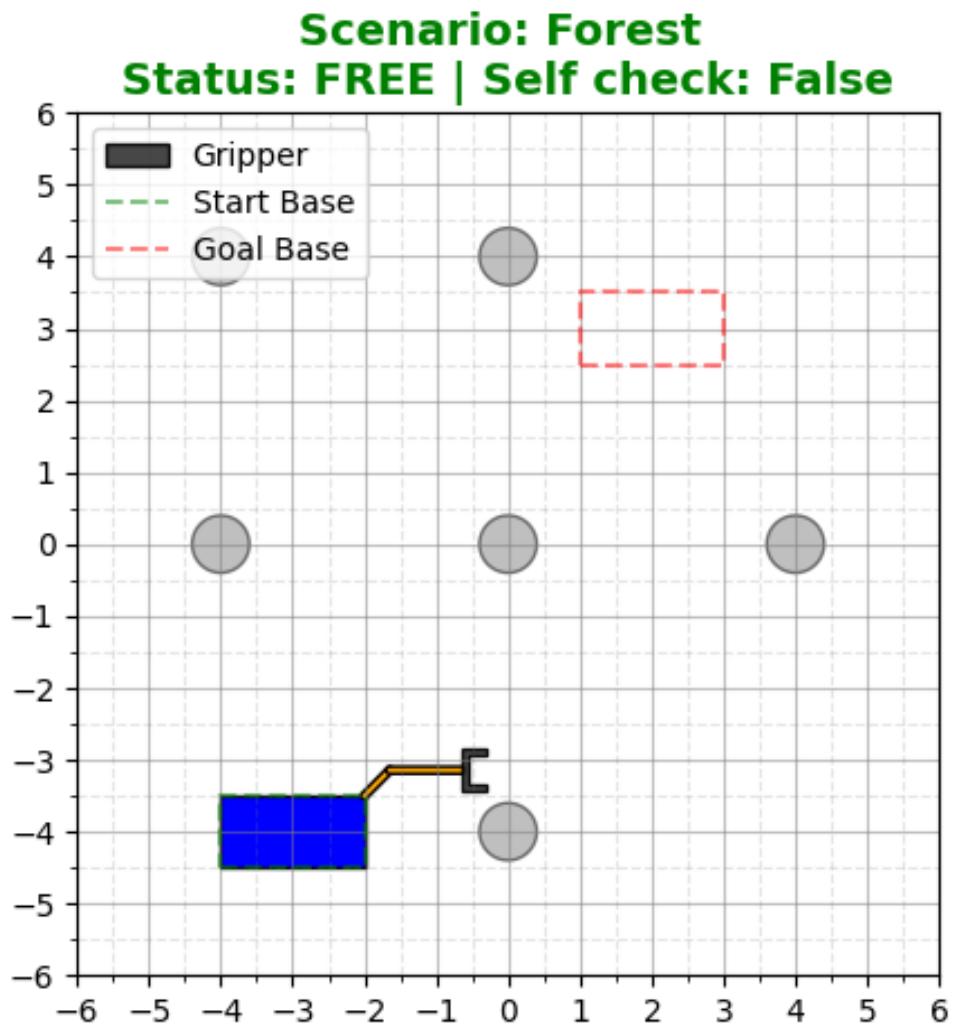
Mechatronik



Mobiler Manipulator

Aufgabe

- Mobiler Manipulator mit Arm
 - Bewegliche Basis (3 Freiheitsgrade)
 - Arm mit 2 rotatorischen Gelenken (2 Freiheitsgrade)
- Teilbereiche:
 - Implementierung eines CollisionCheckers für einen planaren Roboter (Basis + 2 Links).
 - Benchmarking verschiedener Planungsverfahren (LazyPRM vs. VisibilityPRM) in unterschiedlichen Szenarien.
 - Durchführung einer Pick-and-Place Aufgabe, bei der ein Hindernis gegriffen und transportiert wird.
 - Theoretische Betrachtung von Systemerweiterungen (translatorische Gelenke und Pfadglättung).



Geometrische Definition & Konfigurationsraum

- Mobile Basis:

```
ROBOT_BASE_SHAPE = [(0, 0), (2, 0), (2, 1), (0, 1)]  
ROBOT_CENTER = [1, 0.5]
```

- Roboterarm:

```
ROBOT_ARM_CONFIG = [  
    [0.5, 0.1, [0, 3.14]], # Gelenk 1  
    [1.0, 0.1, [-3.14, 3.14]] # Gelenk 2  
]
```

- Greifer:

```
GRIPPER_SHAPE = [[0.35, 0.3], [0.0, 0.3], [0.0, -0.3], [0.35, -0.3],  
    [0.35, -0.2], [0.1, -0.2], [0.1, 0.2], [0.35, 0.2]]  
GRIPPER_LEN = 0.1
```

- Arm Montage:

```
ARM_OFFSET = (2, 1)
```

- Freiheitsgrade:

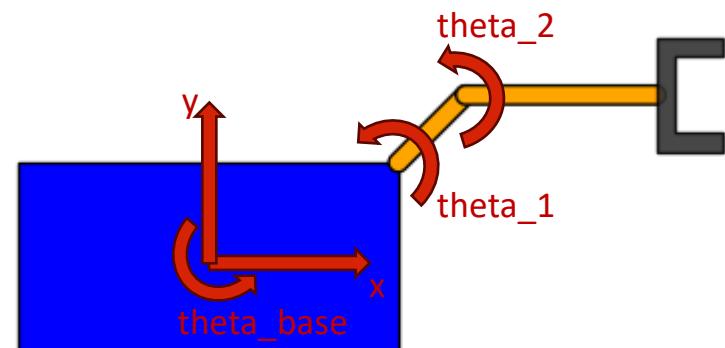
- Basis: x, y, θ_{base}
- Arm: θ_1, θ_2

- 5-dimensionaler Konfigurationsvektor:

- $q = [x \ y \ \theta_{base} \ \theta_1 \ \theta_2]$

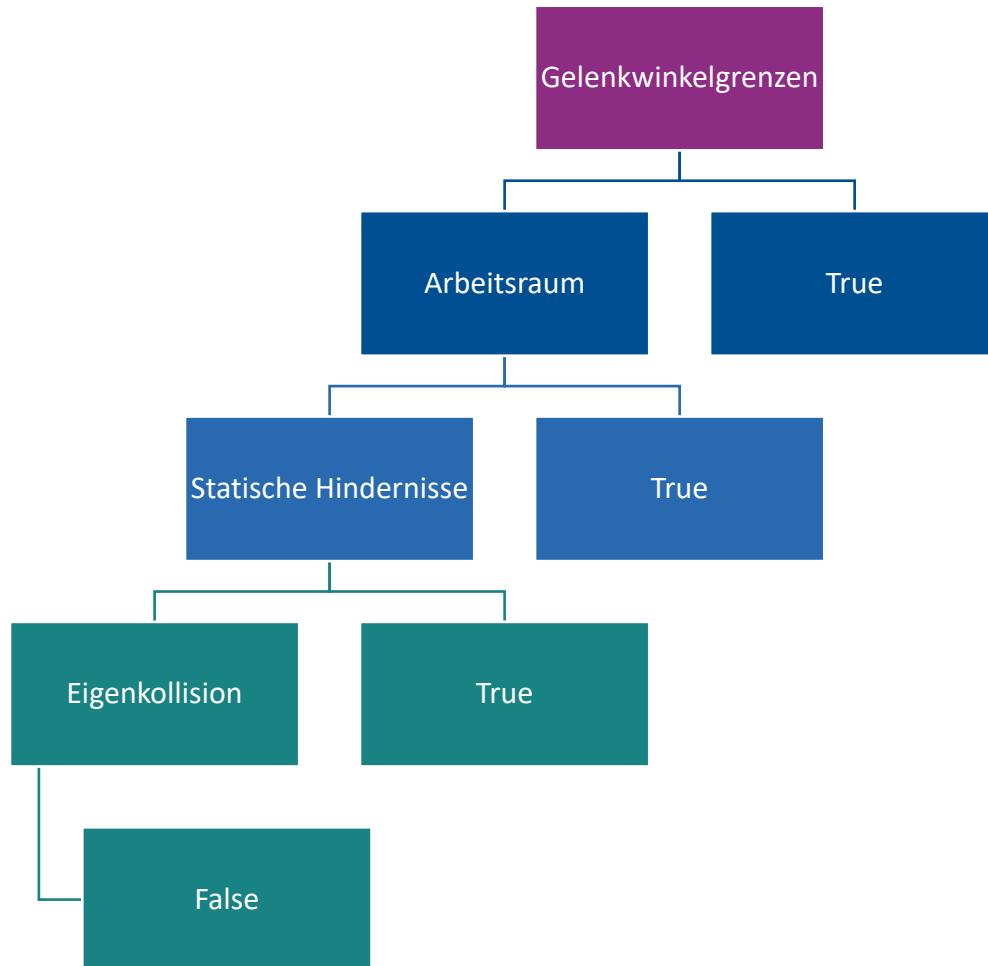
- Startposition:

```
START = [[-3.0, -4.0, 0.0, np.pi/4, -np.pi/4]]
```



Implementierung Collision Checker

Kollisionsprüfung

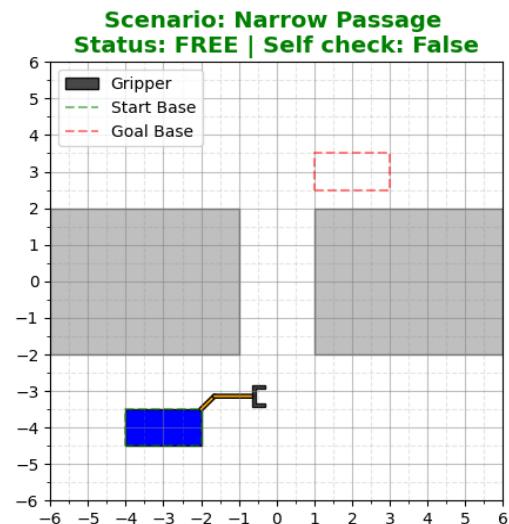
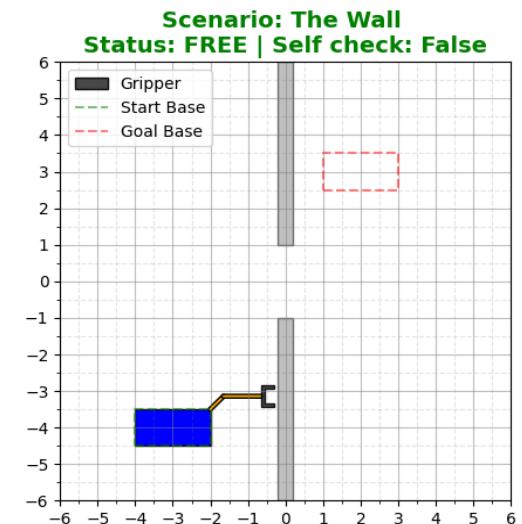


- Gelenkwinkelgrenzen:
 - θ_1, θ_2 in Min / Max-Bereich
- Arbeitsraum:
 - alle Segmente in Weltlimits
- Statische Hindernisse:
 - alle Segmente gegen Liste der statischen Hindernisse
- Eigenkollision:
 - Armsegmente 1 bis n gegen Basis (intersect limit)
 - Greifer gegen Basis
 - Armsegmente 1 bis n-1 gegen Greifer
 - Armsegmente 1 bis n-1 gegen gegriffenes Objekt

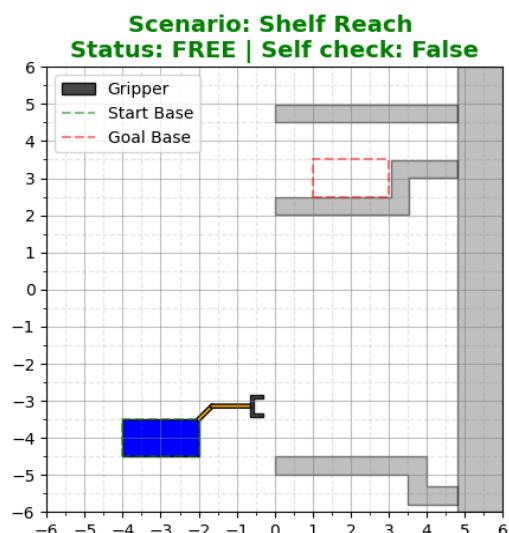
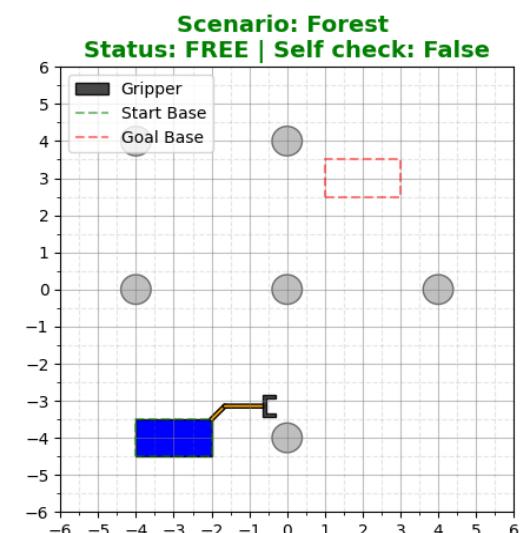
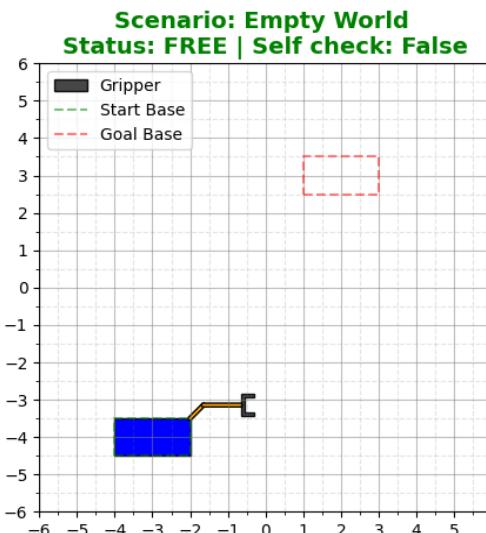
Benchmarking der Planungsverfahren

Aufbau Testumgebung

Planer	Parameter	Wert	Beschreibung
VisibilityPRM	Ntry	500	Sampling-Versuche für Guards
	Initial Roadmap Size	500	Hohe initiale Abdeckung („Flooding“)
	Update Roadmap Size	200	Knoten pro Iteration bei Misserfolg
LazyPRM	k-Nearest	12	Erhöhte Konnektivität für 5D-Raum
	Max. Iterations	15	Obergrenze für Graphen-Wachstum

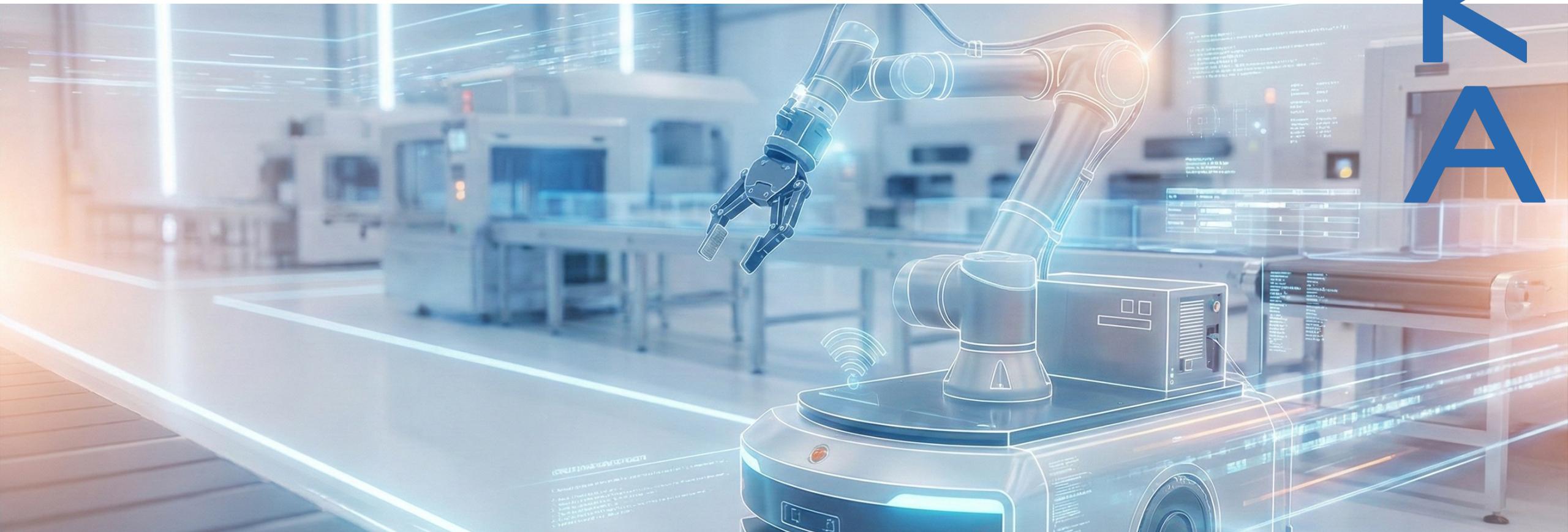


- Pro Szenario und Planer 10 Durchläufe
- Abbruchkriterien:
 - Finden eines Pfades
 - Erreichen der maximalen Knotenzahl

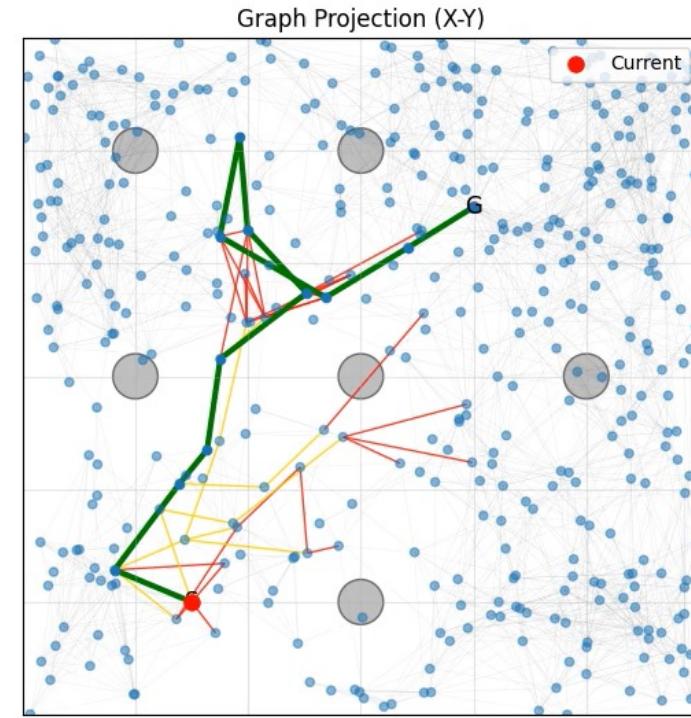
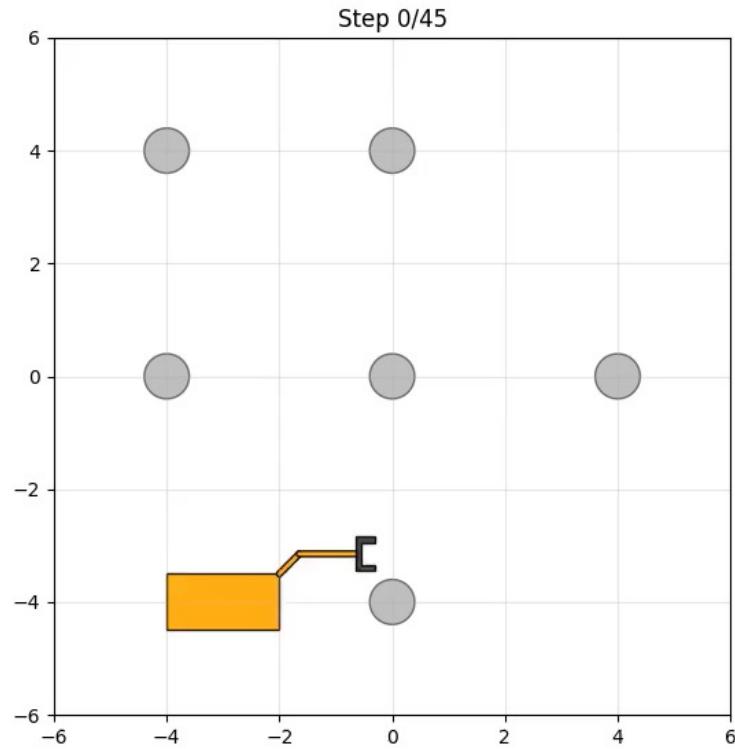




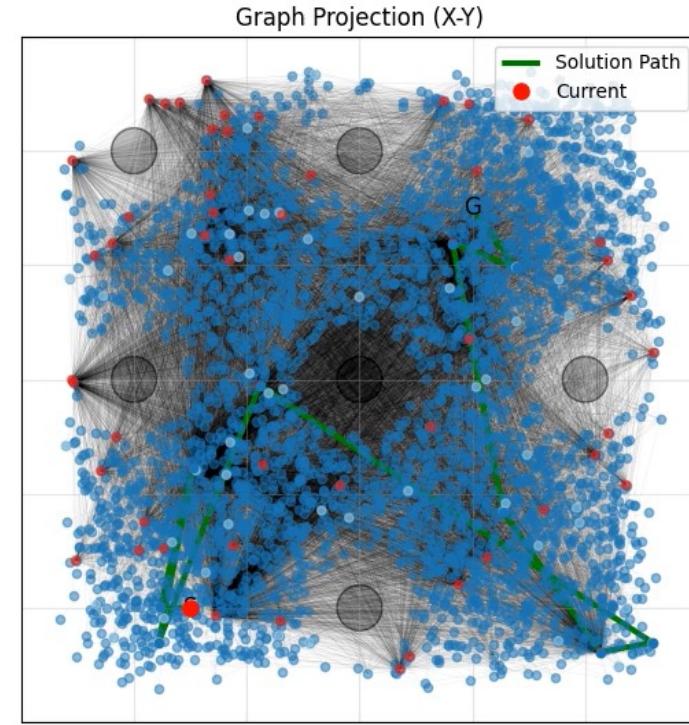
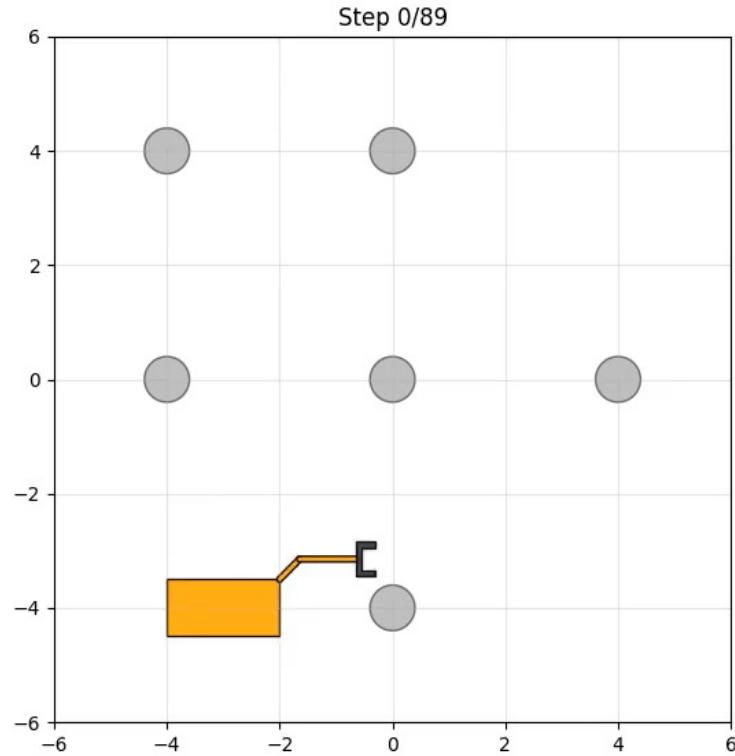
Benchmarking der Planungsverfahren



Qualitative Analyse der Roadmaps - Forest - lazyPRM

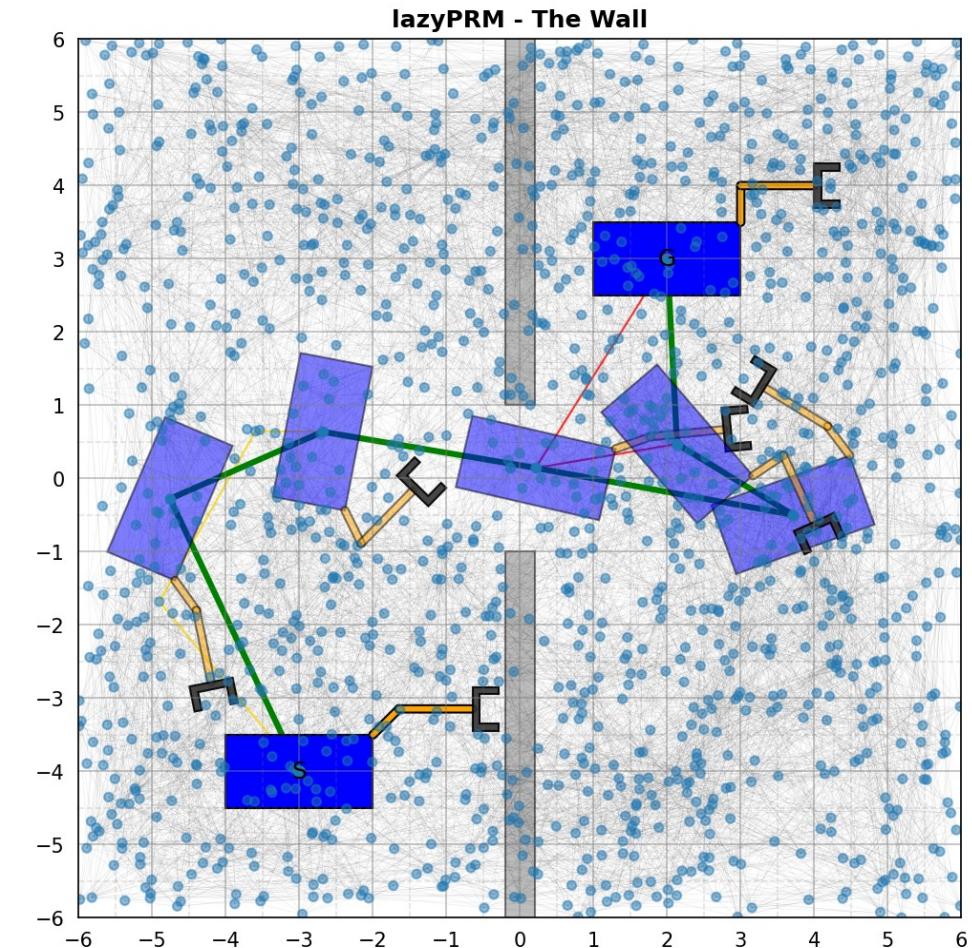
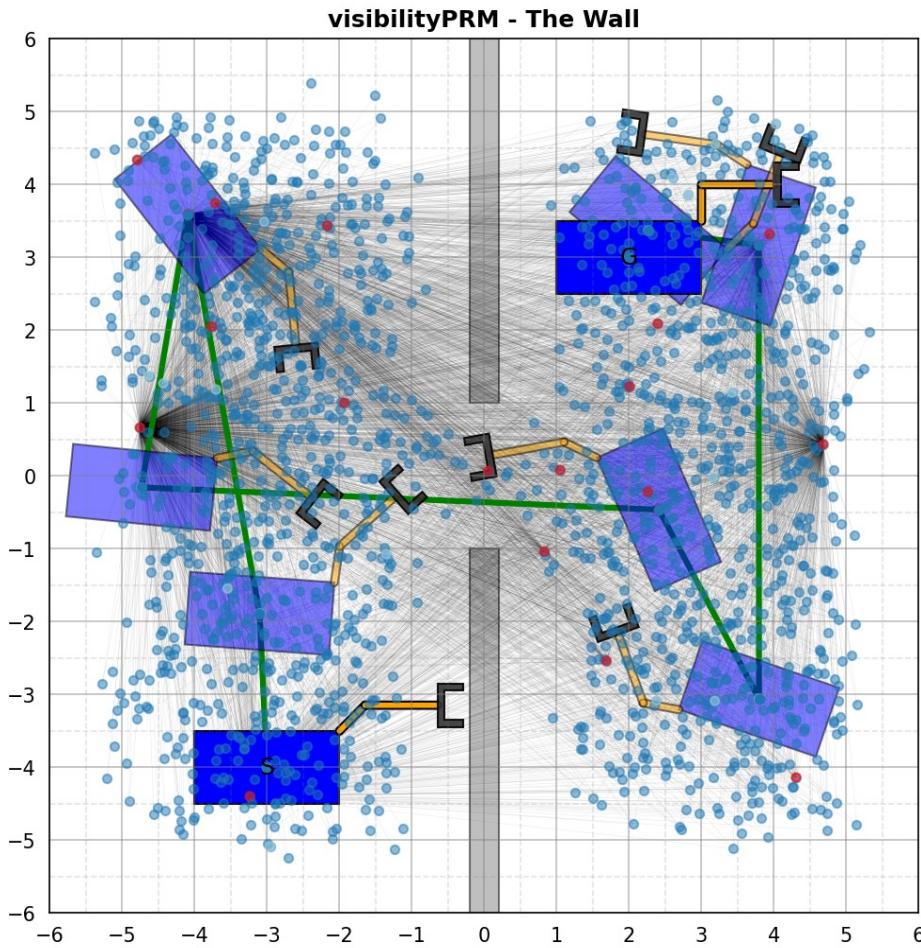


Qualitative Analyse der Roadmaps - Forest - visibilityPRM



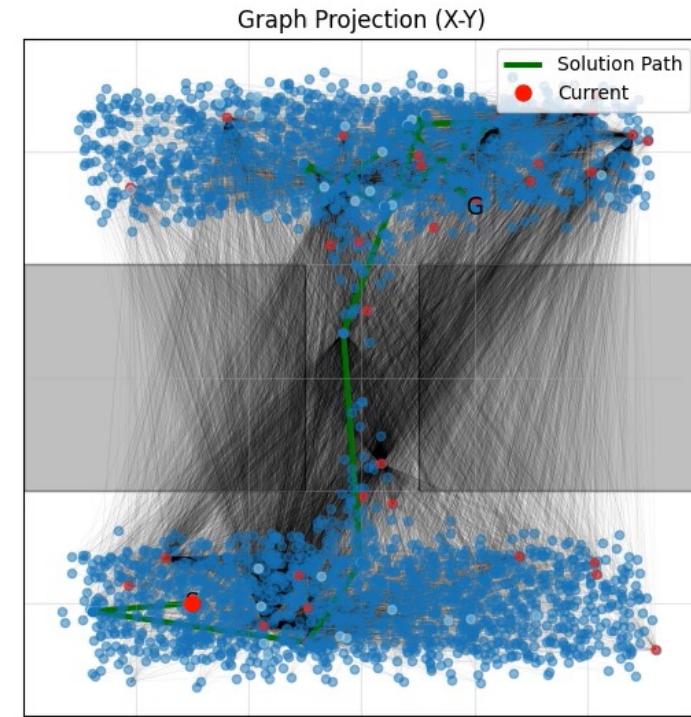
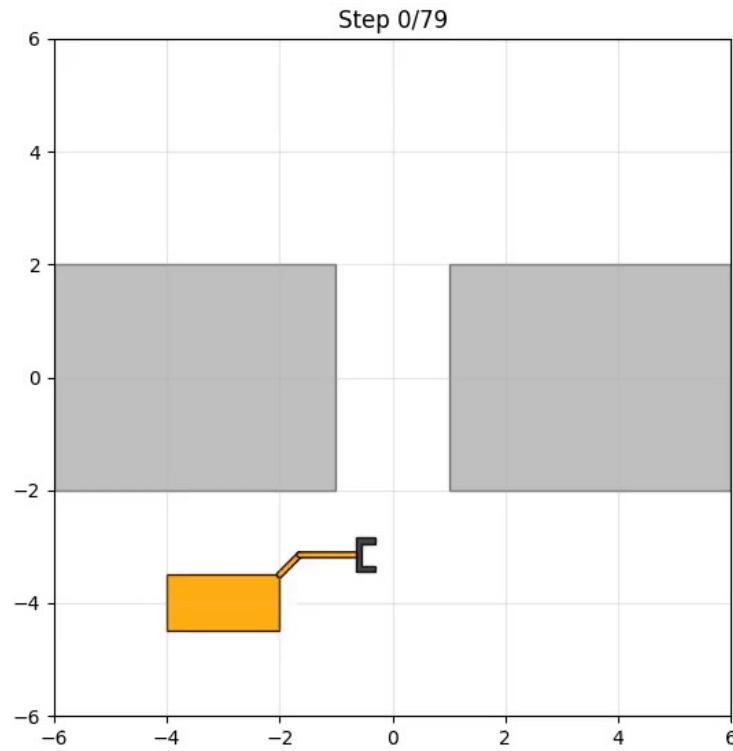
Benchmarking der Planungsverfahren

Qualitative Analyse der Roadmaps – The Wall



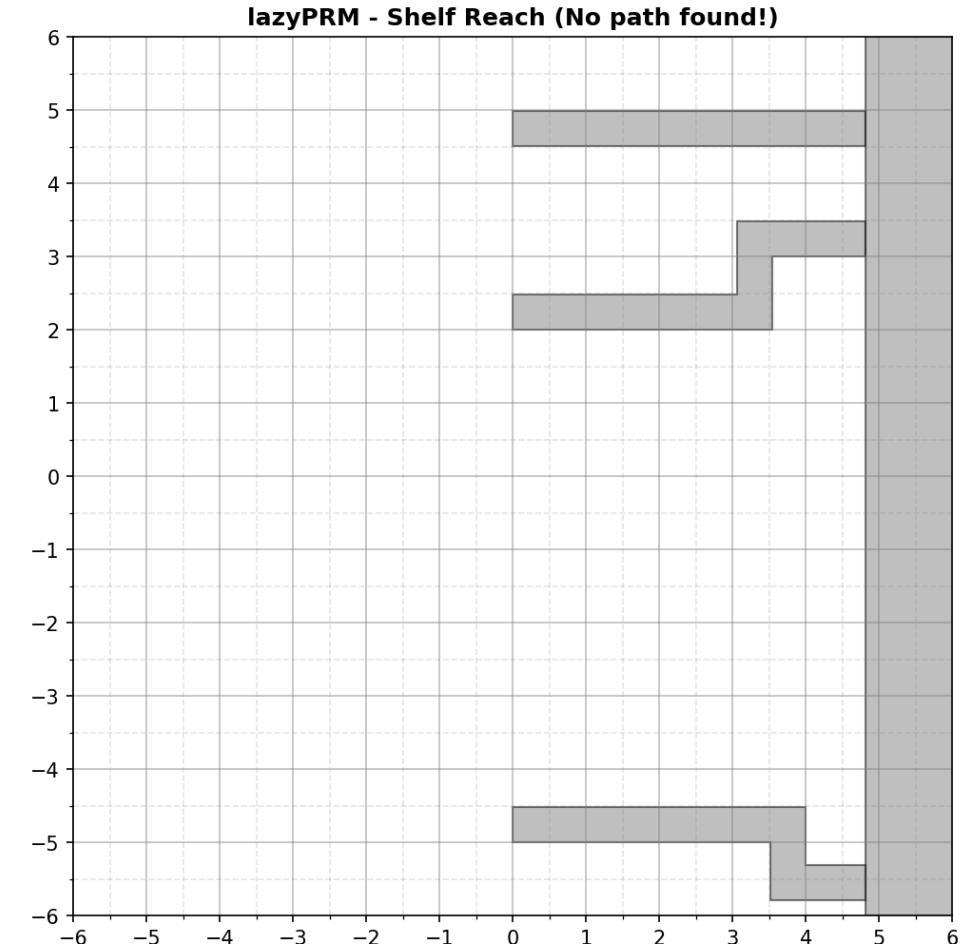
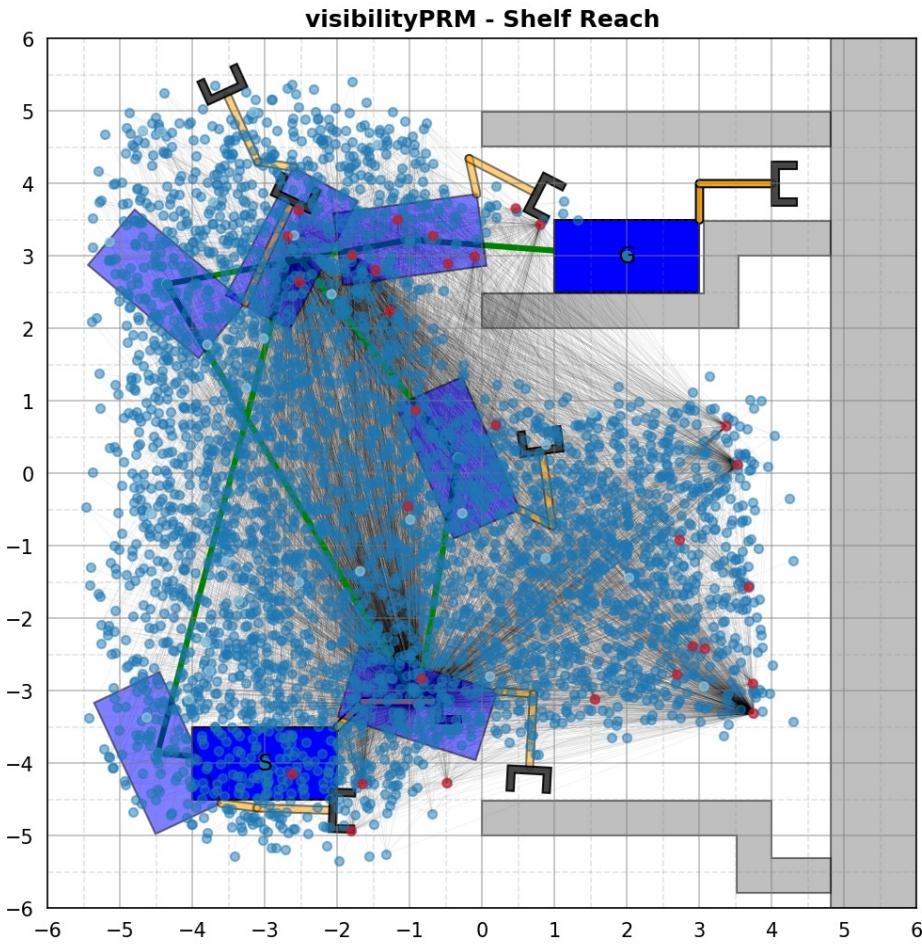
Benchmarking der Planungsverfahren

Qualitative Analyse der Roadmaps – Narrow Passage



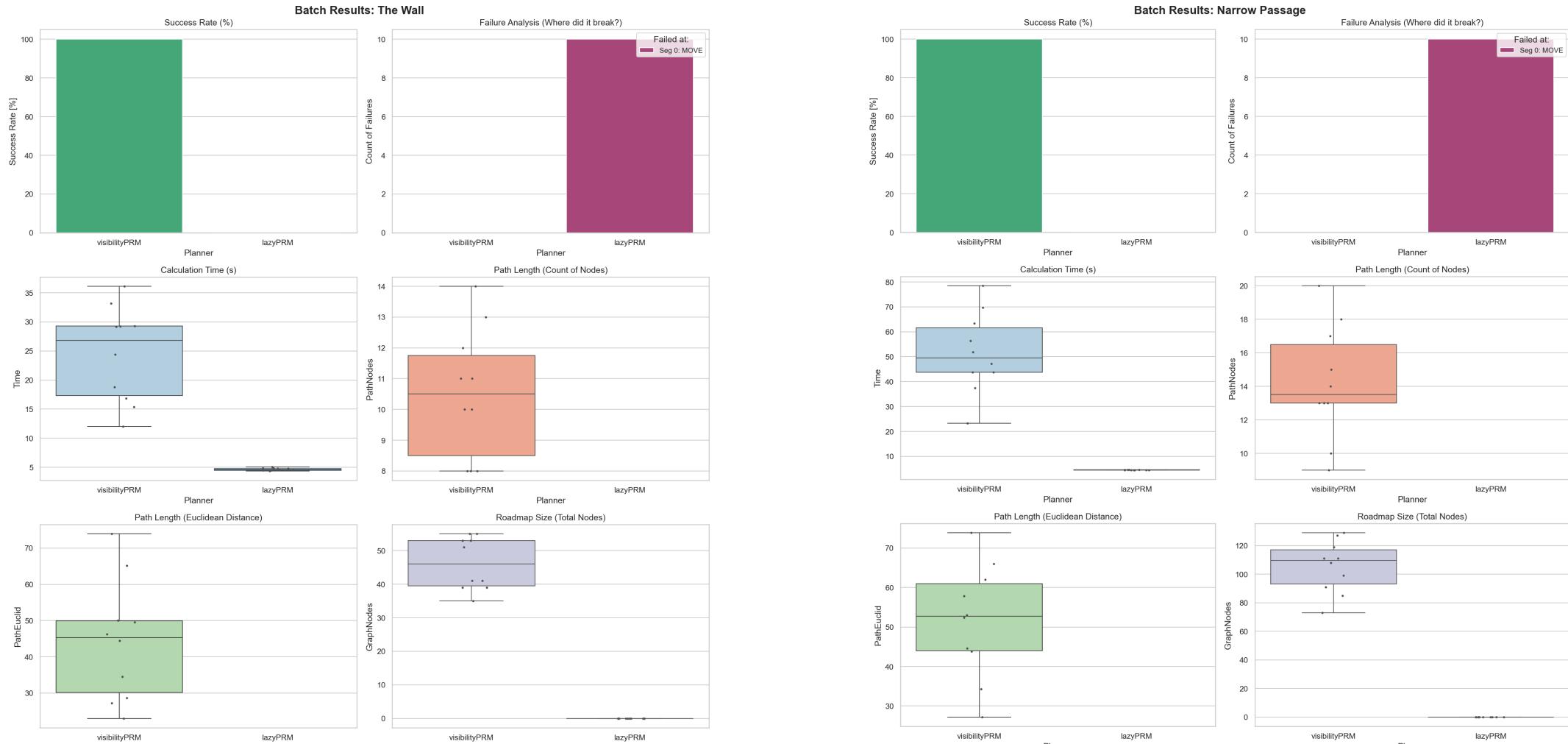
Benchmarking der Planungsverfahren

Qualitative Analyse der Roadmaps – Shelf Reach



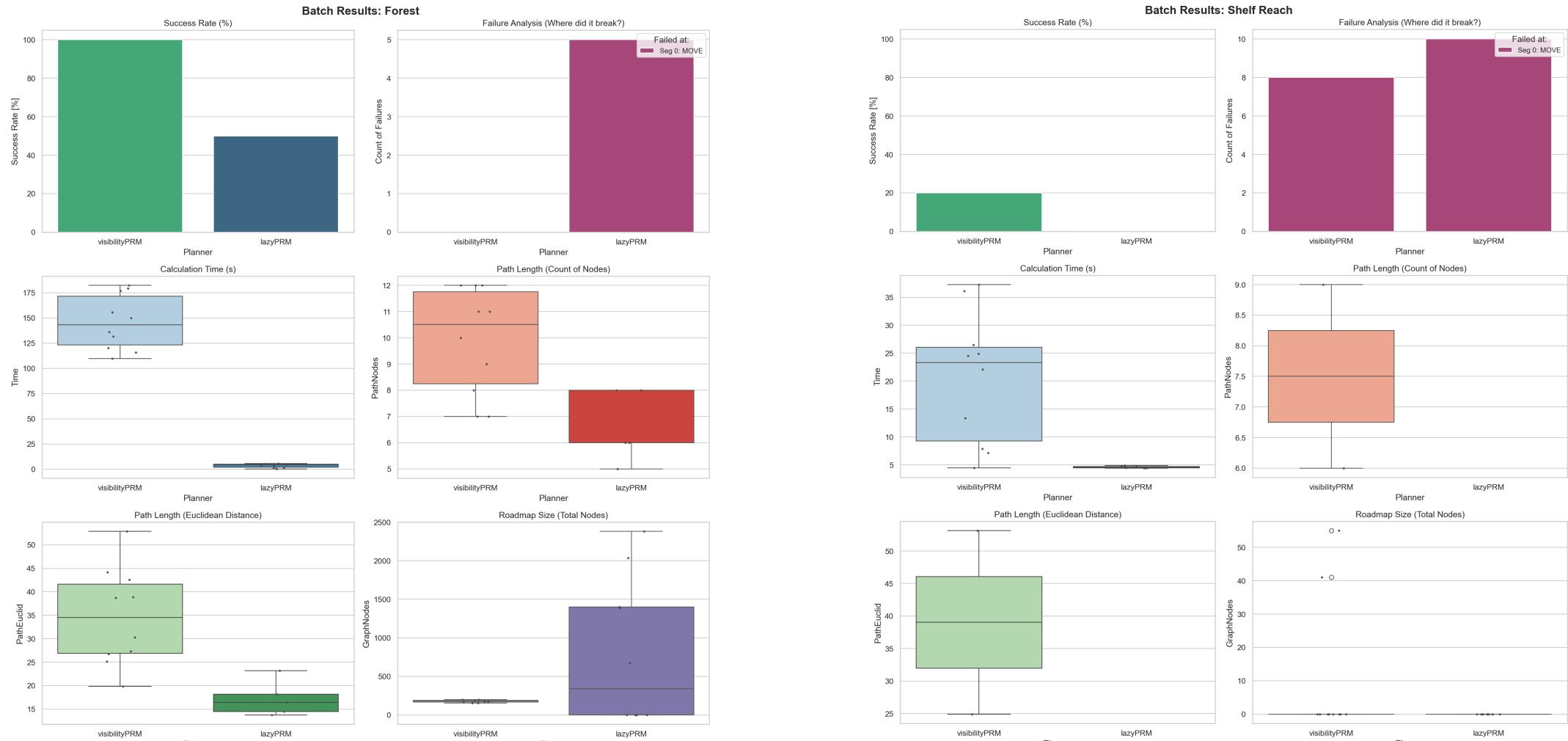
Benchmarking der Planungsverfahren

Quantitative Ergebnisse – mit Eigenkollisionsprüfung



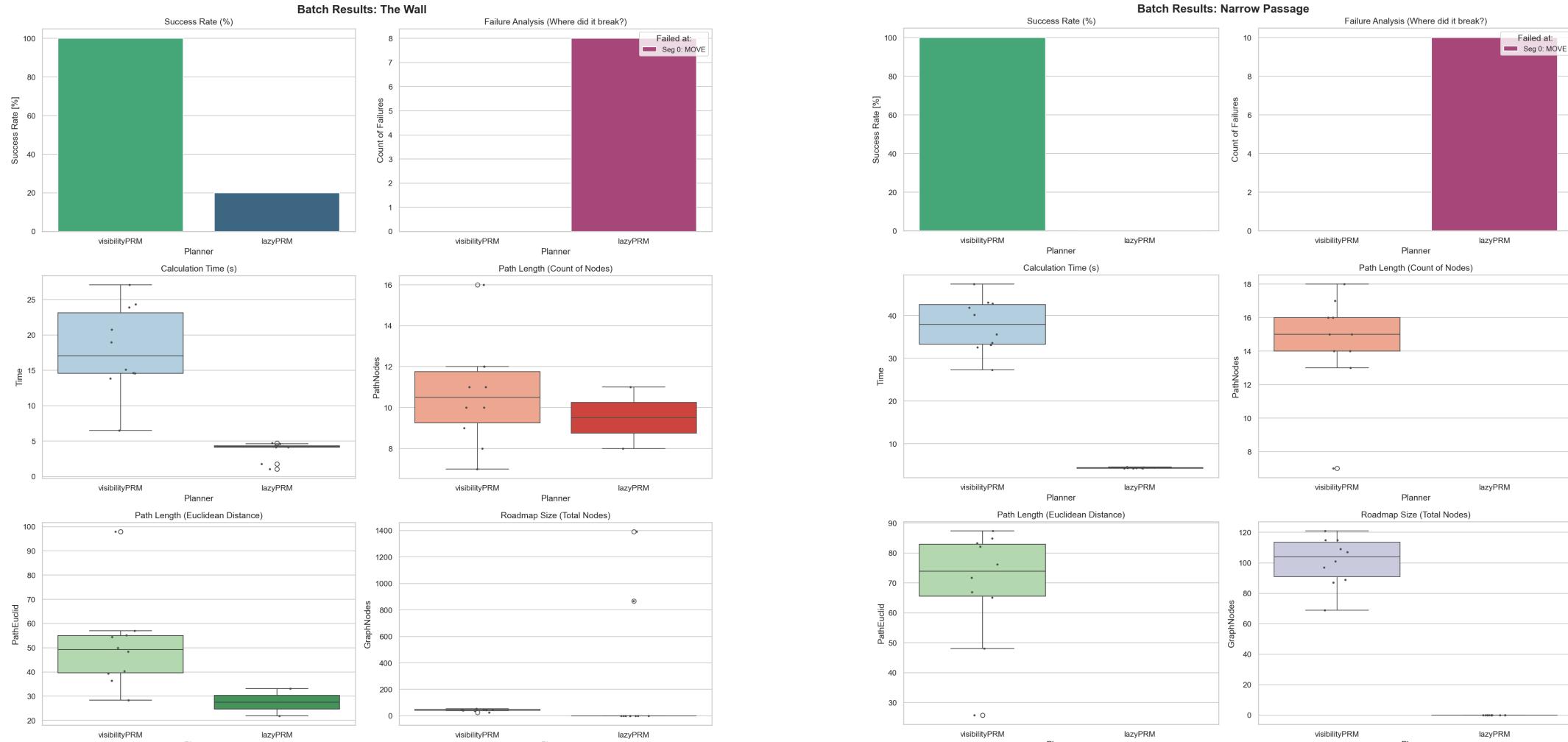
Benchmarking der Planungsverfahren

Quantitative Ergebnisse – mit Eigenkollisionsprüfung



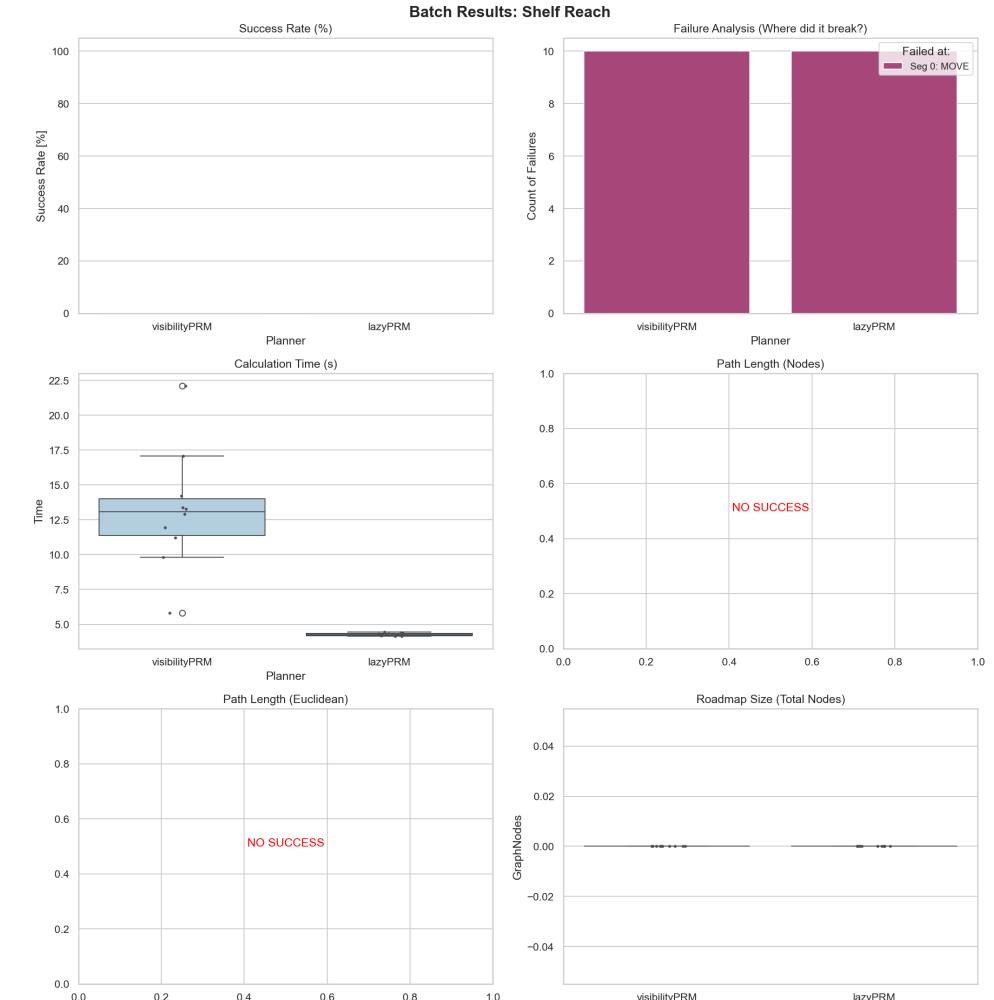
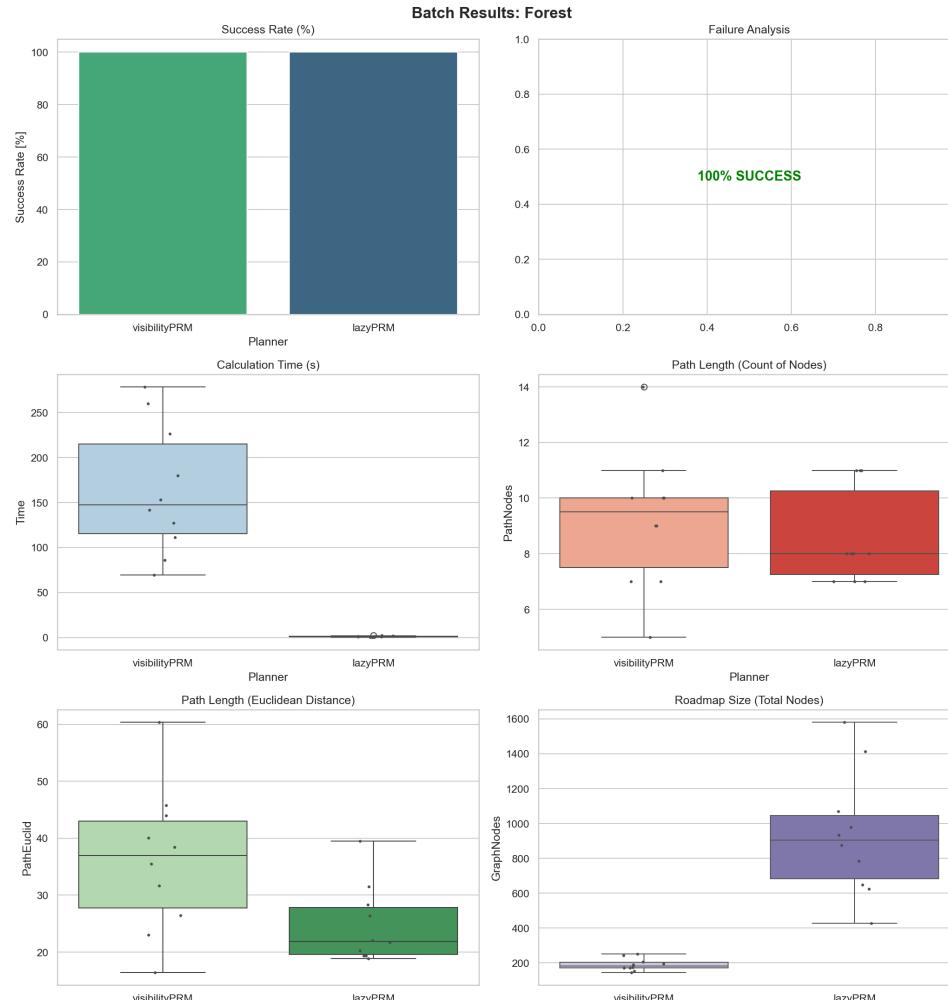
Benchmarking der Planungsverfahren

Quantitative Ergebnisse – ohne Eigenkollisionsprüfung



Benchmarking der Planungsverfahren

Quantitative Ergebnisse – ohne Eigenkollisionsprüfung

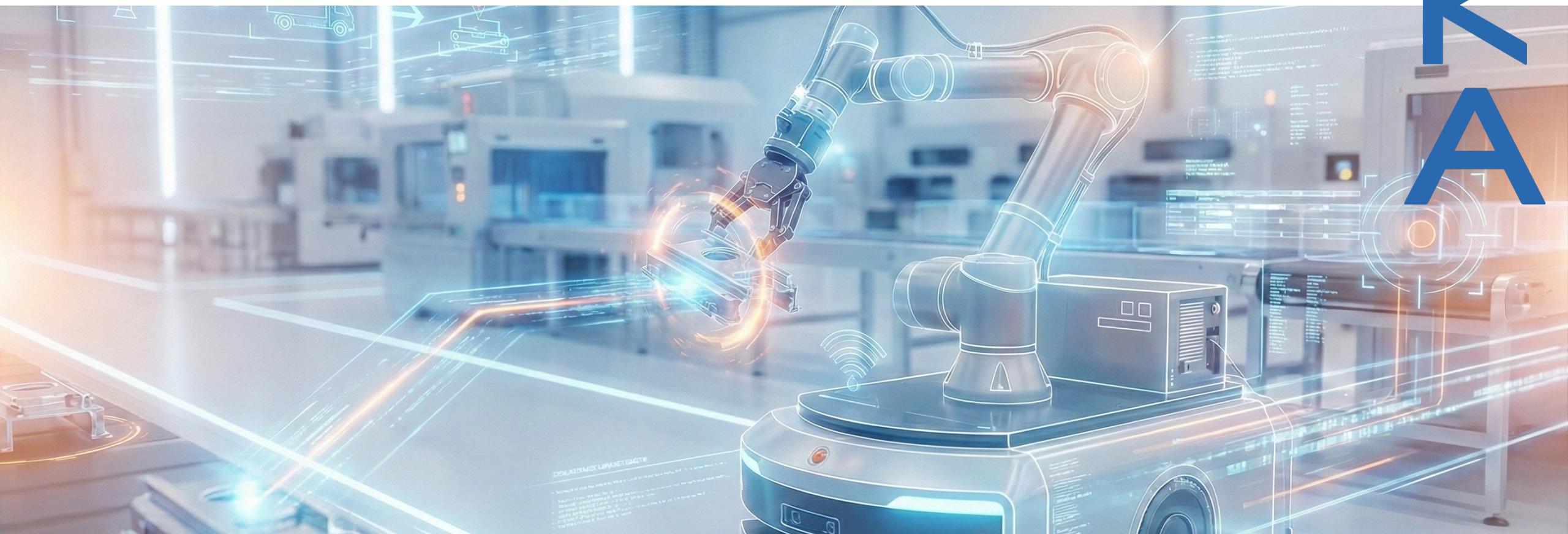


Ergebnisse

- Auswirkung auf die Rechenzeit
 - keine signifikante Reduktion der Rechenzeit
 - dominierender Faktor: Anzahl der Sichtbarkeitsprüfungen (Raycasts) gegen die externen Hindernisse
- VisibilityPRM ist bessere Wahl (bei betrachtetem Manipulator)
 - insbesondere bei unübersichtlichen Szenarien
 - höhere Rechenzeiten vs. Zuverlässigkeit
- LazyPRM besser bei offenen Arealen / Szenarien mit geringer Hindernissdichte



Pick-And-Place



Problemzerlegung und Sequenzierung

- Sampling-basierte Planer nur für Punkt-zu-Punkt Bewegungen
 - Zerlegung in Teilprobleme
 - Erweiterung des 5-dimensionalen Konfigurationsvektors
 - Zusammenfassung der Teilprobleme in Matrix:
- Graph wird für jedes Teilproblem neu aufgebaut
 - Robotergeometrie ändert sich
 - Hindernisliste ändert sich

▪
$$Q = \begin{bmatrix} q_1 & Aufgabe_1 & q_{standoff_n} \\ \vdots & \vdots & \vdots \\ q_n & Aufgabe_n & q_{standoff_n} \end{bmatrix}$$

▪ Mit $q_n = [x_n \ y_n \ \theta_{base_n} \ \theta_{1n} \ \theta_{2n}]$

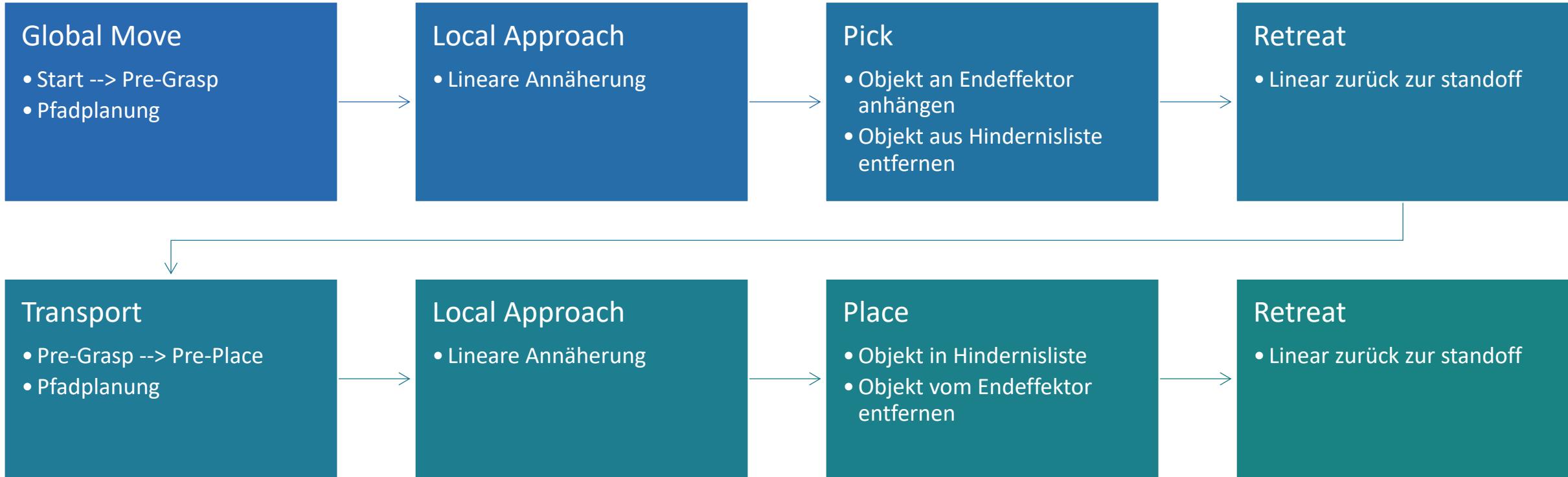
▪ Und $Aufgabe_n \in \{"Pick", "Place", "Move"\}$

▪ Und $q_{standoff_n} = [dx_{standoff_n} \ dy_{standoff_n}]$

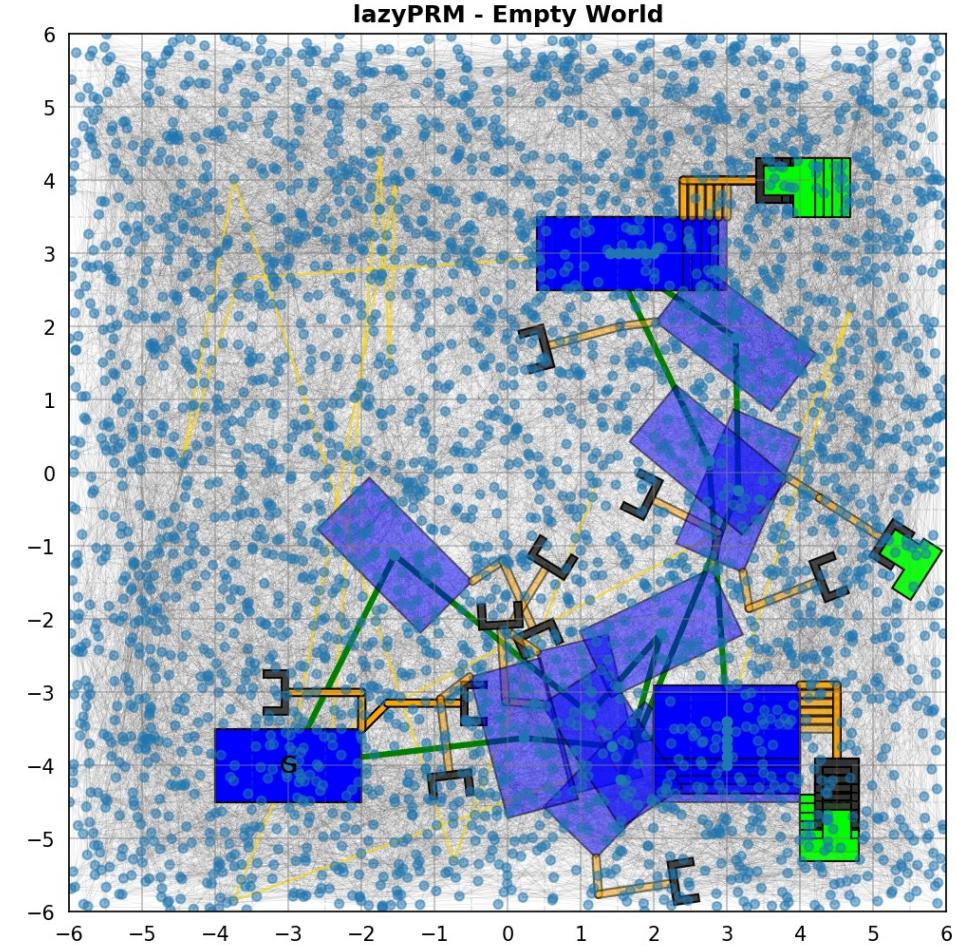
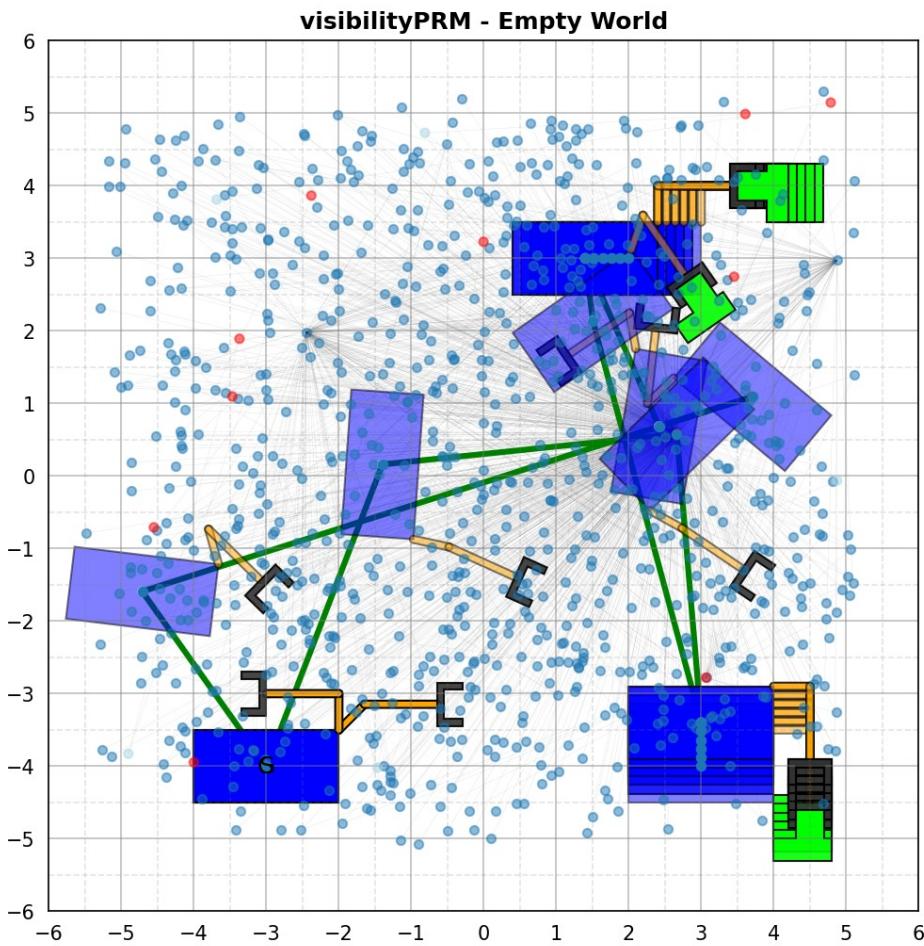
▪ Folglich $[x_{standoff_n} \ y_{standoff_n}] = [x_n \ y_n] - R(\theta_n)q_{standoff_n}$

▪ Mit $R(\theta_n) = \begin{bmatrix} \cos \theta_n & -\sin \theta_n \\ \sin \theta_n & \cos \theta_n \end{bmatrix}$

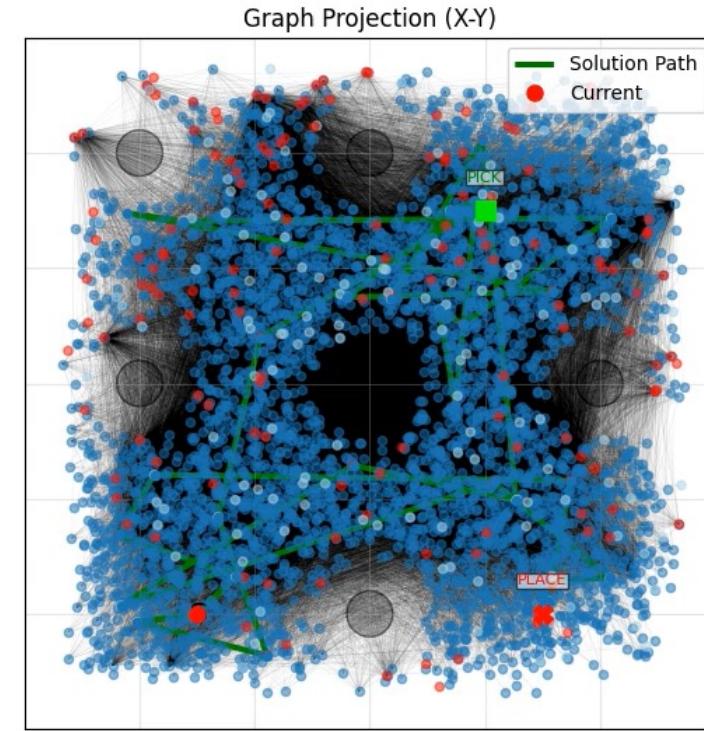
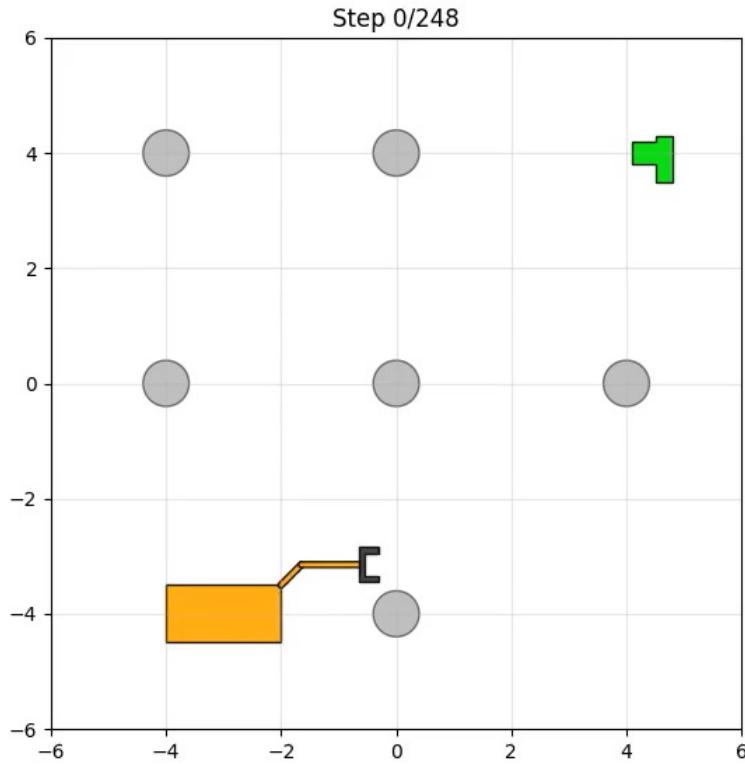
Problemzerlegung und Sequenzierung



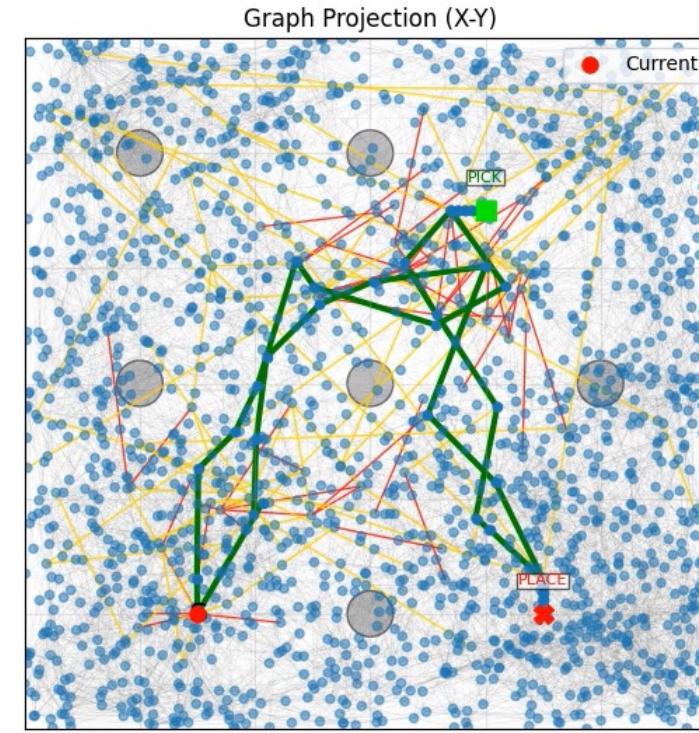
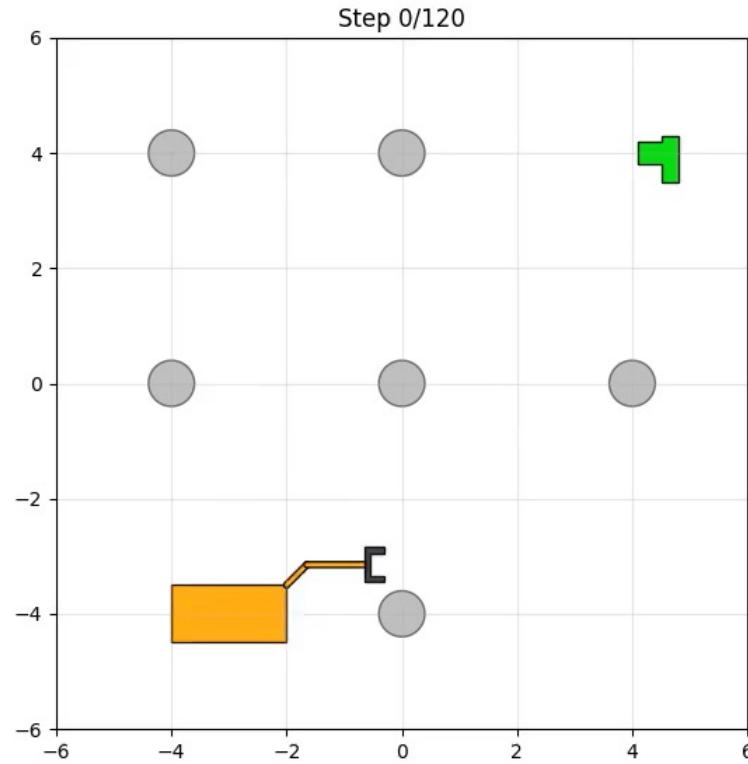
Qualitative Analyse der Roadmaps – Empty World



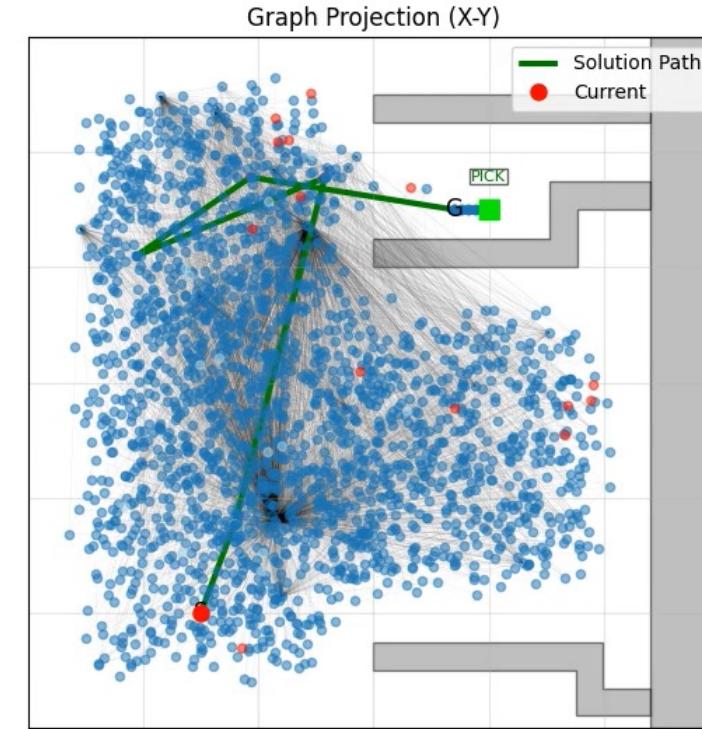
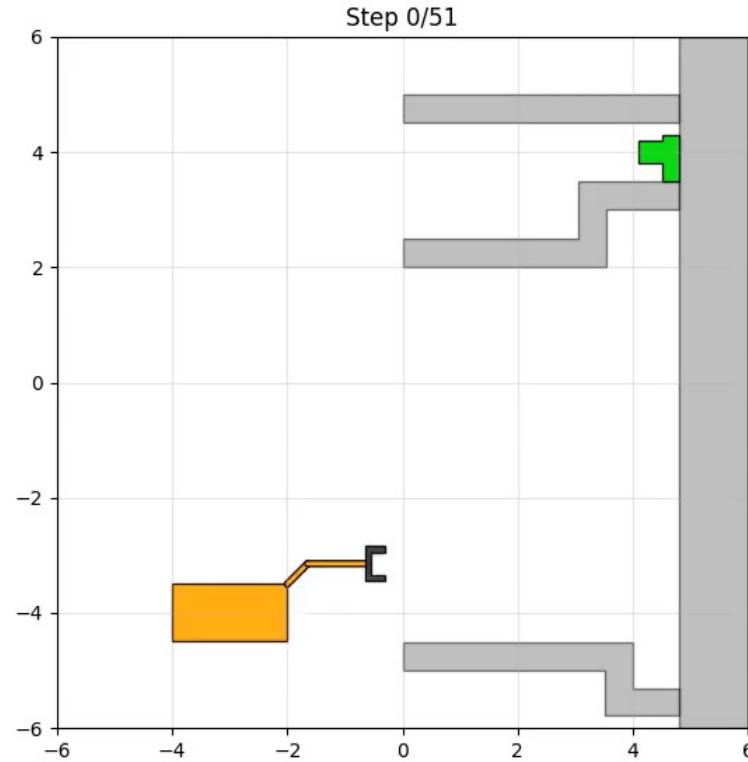
Qualitative Analyse der Roadmaps – Forest – VisibilityPRM



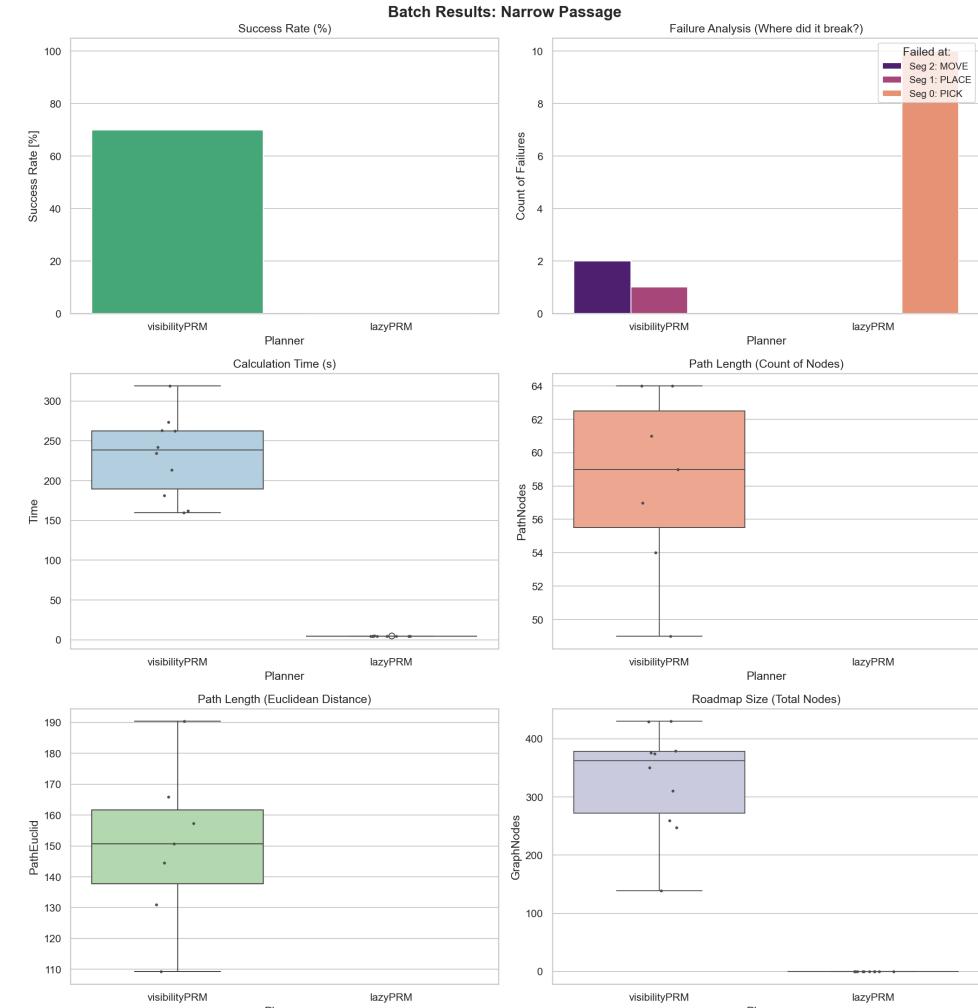
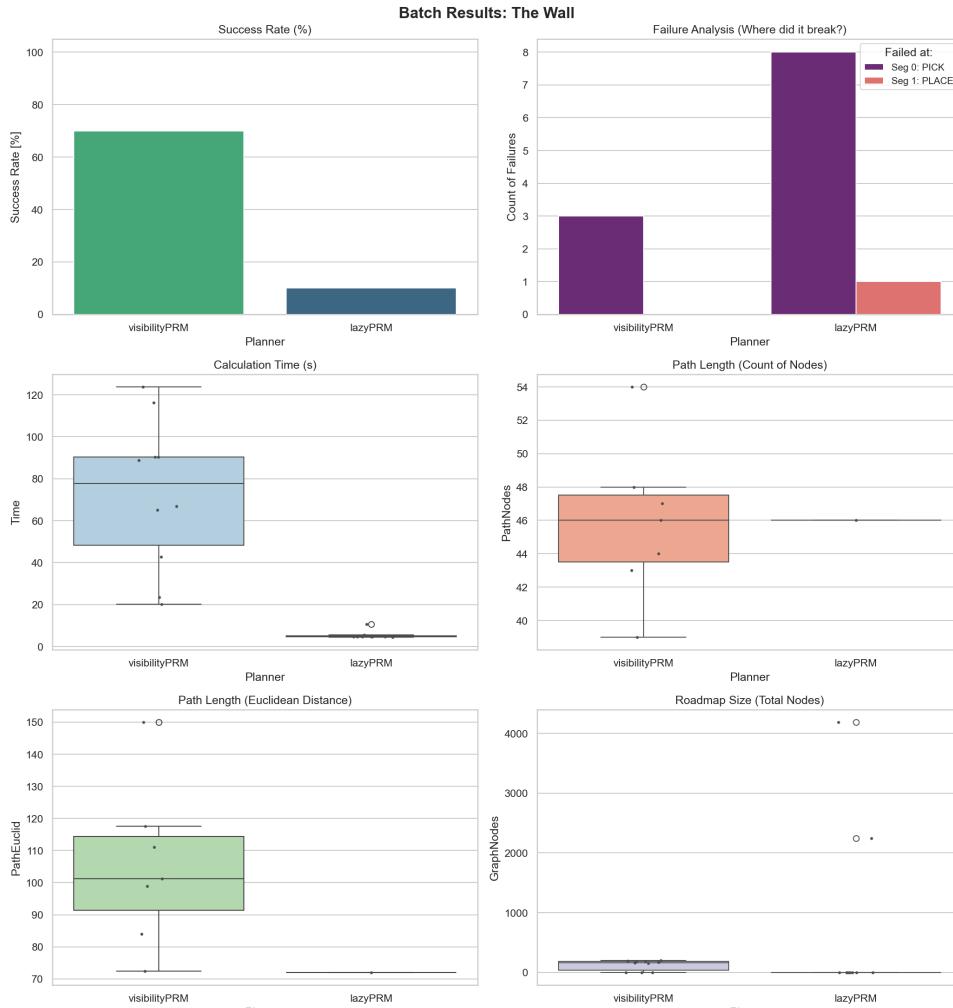
Qualitative Analyse der Roadmaps – Forest – LazyPRM



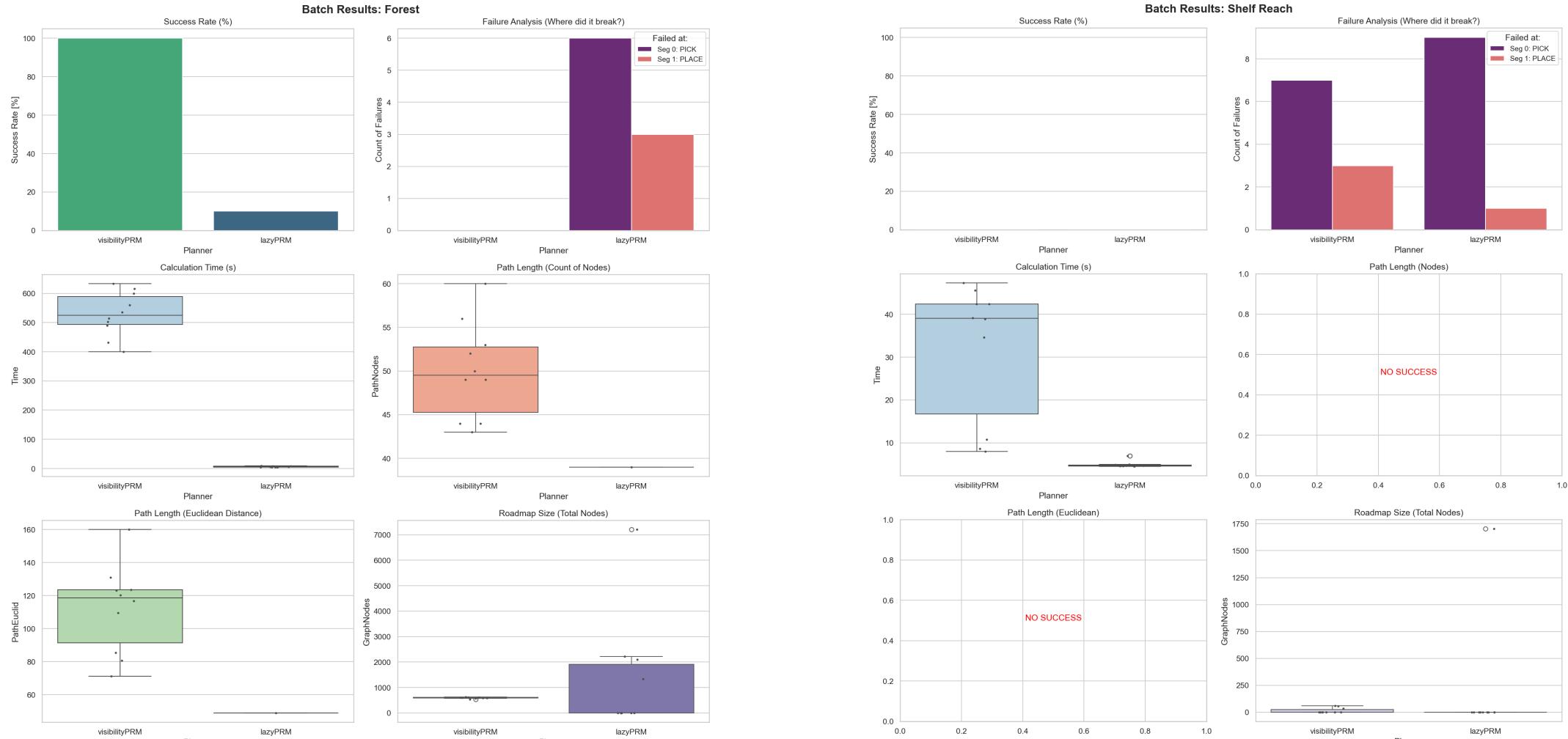
Qualitative Analyse der Roadmaps – Shelf Reach – VisibilityPRM



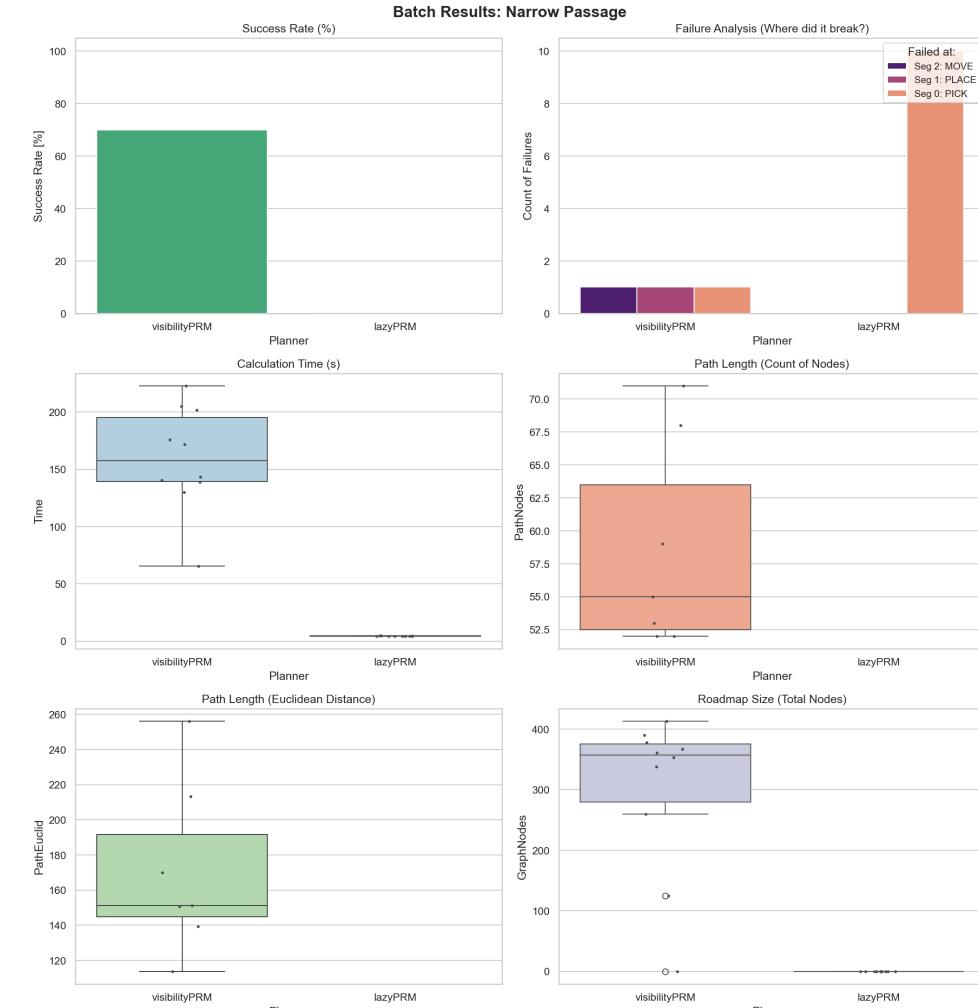
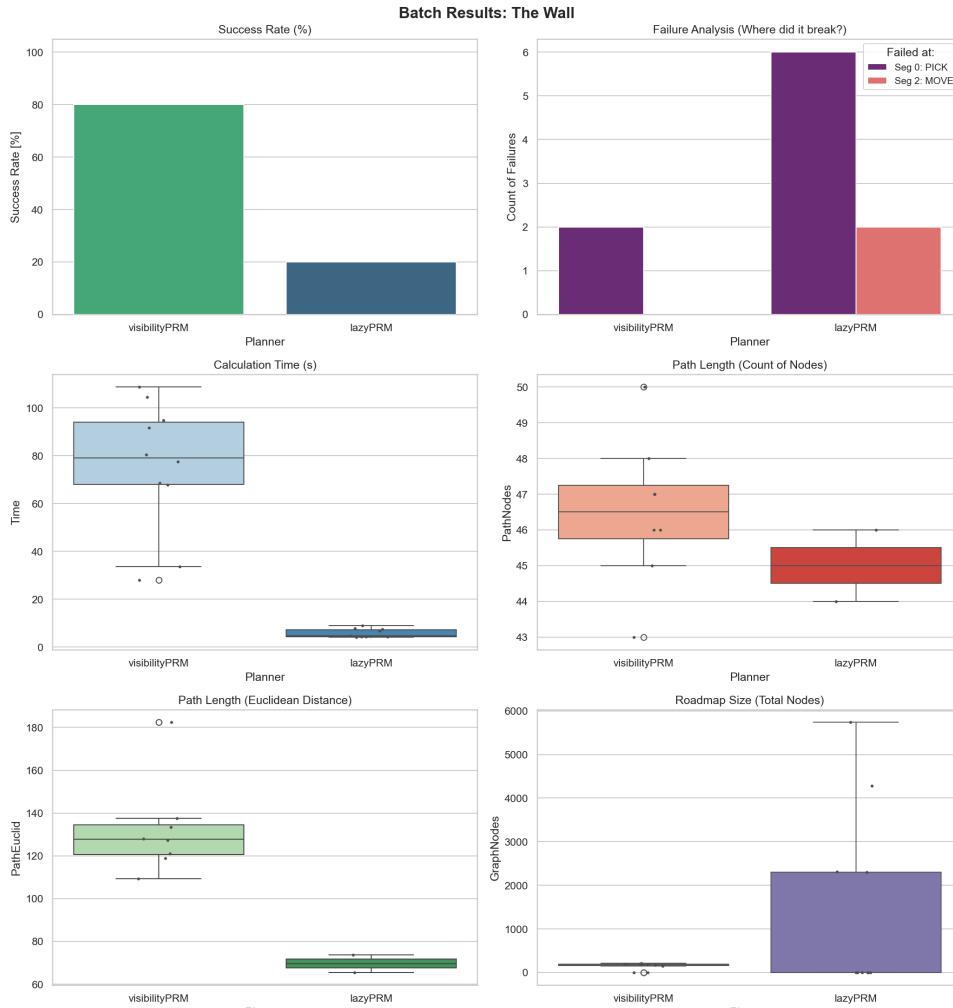
Quantitative Ergebnisse – mit Eigenkollisionsprüfung



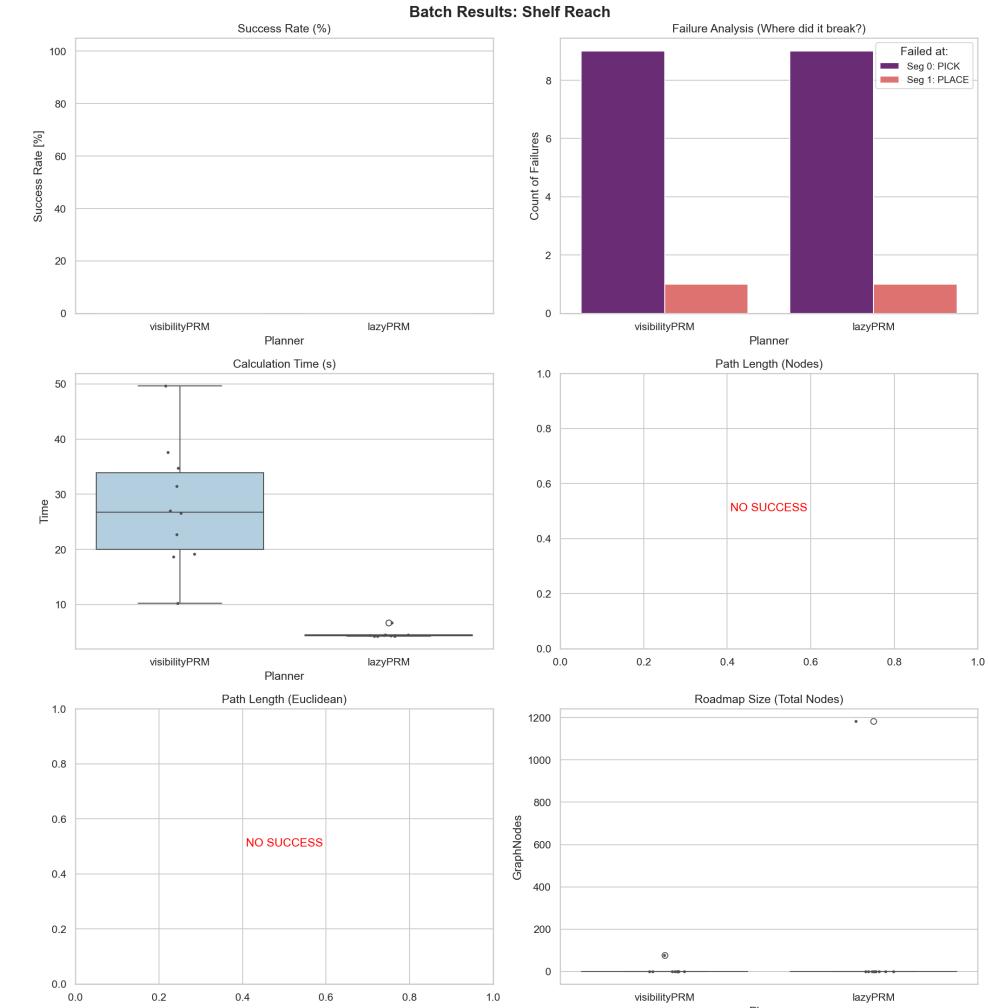
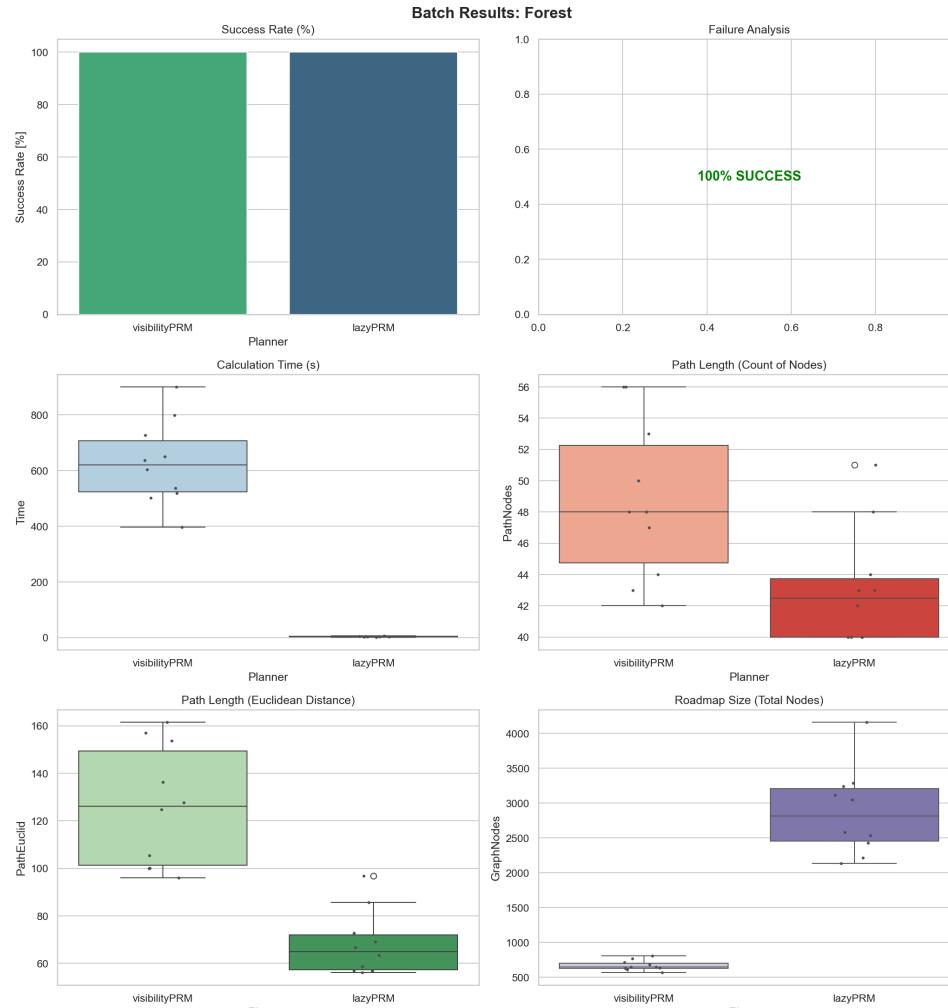
Quantitative Ergebnisse – mit Eigenkollisionsprüfung



Quantitative Ergebnisse – ohne Eigenkollisionsprüfung



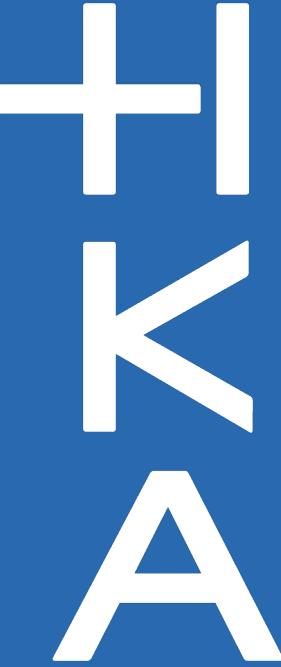
Quantitative Ergebnisse – ohne Eigenkollisionsprüfung



Pick-And-Place

Ergebnisse

- Erhöhte Komplexität
 - Signifikant schwieriger als reine Navigation durch geometrische Restriktionen des Objekts
- Vergleich der Planer (Robustheit vs. Effizienz)
 - LazyPRM: Hohe Effizienz in offenen Bereichen (1–2 s), aber hohe Fehlschlagrate in komplexen Szenarien
 - VisibilityPRM: Hohe Robustheit in engen Umgebungen (Guard-Nodes), jedoch rechenintensiv (bis zu 600 s im Forest-Szenario)
- Fehlerursachen & Grenzen
 - Fehlerquellen: Variieren je nach Szenario (Eigenkollision im Forest vs. Umweltgeometrie in Narrow Passage)
 - Systemgrenzen: Shelf Reach-Szenario zeigt Notwendigkeit für spezialisierte Heuristiken oder höhere Sampling-Dichte
- Gesamtfazit
 - Bei komplexen Manipulationsaufgaben ist eine intelligente Sampling-Strategie entscheidender als reine Rechengeschwindigkeit



Vielen Dank!