

2D Corridor Membranes and Near-Start Two-Target Bimodality

New Extension Report Bridging Previous 2D Results

Headline Results

This report adds two new research tracks on the 2D rectangular setting while keeping the same phase criteria and peak-valley style used in earlier reports.

1. **Corridor + one target:** replace fully reflecting corridor walls by semi-permeable membranes, with both symmetric ($\kappa_{\uparrow} = \kappa_{\downarrow}$) and asymmetric ($\kappa_{\uparrow} \neq \kappa_{\downarrow}$) cases.
2. **No corridor + two targets:** place one target close to the start and keep the other at the far side, to test robust “fast near-hit + slow far-hit” splitting.

Main outcomes:

- For symmetric membranes, clear-double forms a continuous ultra-low leakage band (see Figure 1 and Table 1); it then enters a weak-double transition band and finally turns mostly single-peak at higher leakage.
- A clear asymmetric representative ($\kappa_{\uparrow} = 0.002, \kappa_{\downarrow} = 0$) keeps robust bimodality and changes valley/second-peak composition in the L×R class decomposition.
- In the no-corridor two-target setup, near-start placement yields stable clear-double: the representative case $(b_x, d, \Delta y) = (0.07, 2, 3)$ gives near/far mass $\approx 0.326/0.674$ with peaks around (11, 739); the dedicated showcase clear-instance is (0.12, 4, 1) with near/far mass $\approx 0.350/0.650$ and a deeper valley (valley/max ≈ 0.007).

1 Model and Shared Baseline

We use the existing rectangular 2D framework with reflecting boundaries:

$$L_x = 60, \quad q = 0.8, \quad x_s = 7, \quad x_t = 58.$$

For the corridor branch we use an updated anchor tuned for higher clear-double coverage:

$$h = 2, \quad m = 5, \quad \delta_{\text{core}} = 1.00, \quad \delta_{\text{open}} = 0.55, \quad b_x = -0.08.$$

1.1 Membrane Definition

Each wall-crossing attempt at a membrane edge is accepted with probability κ and reflected with probability $1 - \kappa$. Symmetric case: $\kappa_{\uparrow} = \kappa_{\downarrow} = \kappa$. Asymmetric case: $\kappa_{\uparrow} \neq \kappa_{\downarrow}$.

1.2 Class Labels for Mechanism

We keep peak/valley windows and use a compact 4-class coding:

- L : membrane leak happened or not;
- R : committor recross happened or not.

Classes are LOR0, LOR1, L1R0, L1R1.

2 Track A: Corridor + One Target + Membranes

2.1 Symmetric Scan

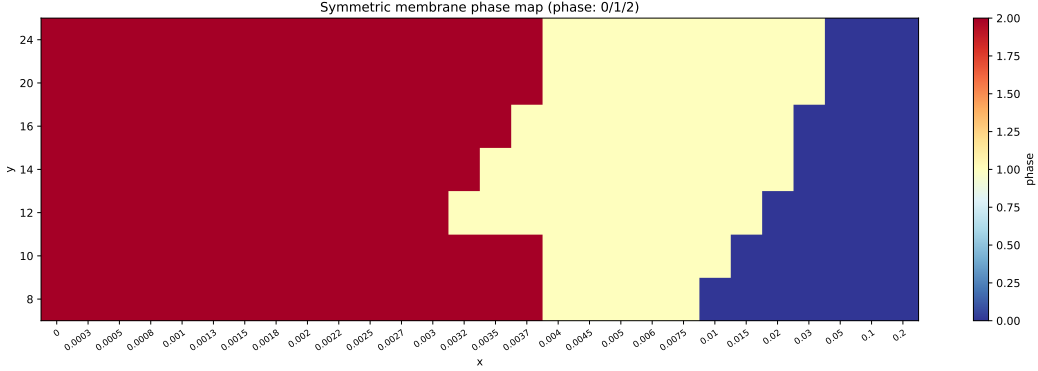


Figure 1: Symmetric membrane phase map (rows: W_y , columns: κ).

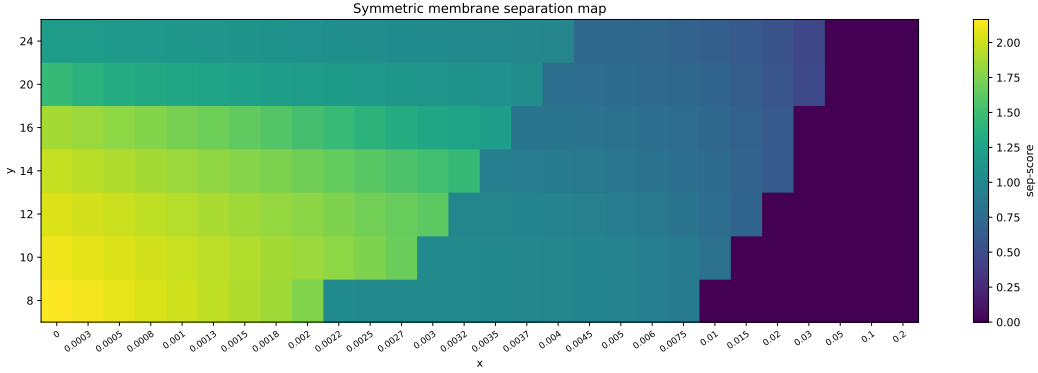


Figure 2: Symmetric membrane separation map (sep-score).

In this run (taking the $W_y = 16$ slice as reference), strong bimodality gradually weakens with permeability: clear-double for $\kappa \in [0, 0.0035]$, weak-double around $\kappa \in [0.0037, 0.02]$, and mostly single-peak for $\kappa \geq 0.03$.

2.2 Asymmetric Scan

Representative overlays:

With strict recross (visit Σ_{q^*} then return to A), the exact decomposition shows: peak1 is LOR0-dominated, valley remains mostly LOR0 with increased L1R0, and peak2 is LOR1-dominated. So the late peak is mainly a commitment-failure effect, not necessarily leak-driven.

3 Track B: No Corridor + Two Targets (Near-Start Target)

We remove corridor geometry and keep reflecting boundaries only. One target is placed near the start, and we optimize in two stages: first maximize clear coverage over candidate grids, then select a dedicated clear-instance from all candidate results for the sharpest double-peak visualization.

Representative case ($W_y = 16$, $b_x = 0.07$, $d = 2$):

The clear-instance metrics (near/far masses, sep-score, valley/max, peak/min margins, showcase score, and splitting-committor closure error $\varepsilon_{\text{split}}$) are listed in Table 6. In this round we apply a robustness gate before final ranking (prioritizing positive **peak-margin/min-margin** with deep-valley preference), so the selected no-corridor case is not only phase=2-valid but also visually stable under threshold perturbations. The current auto-selected clear-instance

κ	phase	t_{p1}	t_{p2}	valley/max	sep-score
0	2	395	1711	0.058	1.87
0.0003	2	395	1710	0.061	1.84
0.0005	2	395	1708	0.063	1.80
0.0008	2	395	1706	0.066	1.76
0.001	2	394	1704	0.069	1.72
0.0013	2	394	1702	0.072	1.68
0.0015	2	394	1700	0.074	1.64
0.0018	2	394	1698	0.077	1.59
0.002	2	394	1696	0.080	1.53
0.0022	2	393	1694	0.082	1.46
0.0025	2	393	1692	0.085	1.40
0.0027	2	393	1690	0.087	1.34
0.003	2	393	1689	0.090	1.29
0.0032	2	393	1687	0.093	1.25
0.0035	2	393	1685	0.095	1.21
0.0037	1	392	1683	0.098	0.84
0.004	1	392	1681	0.100	0.84
0.0045	1	392	1677	0.105	0.82
0.005	1	392	1673	0.110	0.81
0.006	1	391	1664	0.120	0.79
0.0075	1	390	1652	0.134	0.77
0.01	1	388	1629	0.157	0.73
0.015	1	386	1576	0.198	0.67
0.02	1	383	1505	0.235	0.61
0.03	0	-	-	-	0.00
0.05	0	-	-	-	0.00
0.1	0	-	-	-	0.00
0.2	0	-	-	-	0.00

Table 1: Symmetric membrane metrics on the $W_y = 16$ slice.

$\kappa_{\uparrow}$	$\kappa_{\downarrow}$	phase	t_{p1}	t_{p2}	sep-score
0	0.0003	2	395	1711	1.85
0.0003	0	2	395	1710	1.85
0.0005	0	2	395	1710	1.84
0	0.0005	2	395	1710	1.84
0.0008	0	2	395	1709	1.82
0.0005	0.0003	2	395	1709	1.82
0.0003	0.0005	2	395	1709	1.82
0	0.0008	2	395	1709	1.82
0.001	0	2	395	1708	1.80
0.002	0	2	394	1704	1.73

Table 2: Top asymmetric membrane cases (ranked by phase and separation).

Window	Case	L0R0	L0R1	L1R0	L1R1
peak1	sym	0.996	0.000	0.004	0.000
peak1	asym	0.998	0.000	0.002	0.000
valley	sym	0.581	0.116	0.299	0.004
valley	asym	0.687	0.142	0.169	0.002
peak2	sym	0.142	0.742	0.080	0.036
peak2	asym	0.149	0.794	0.040	0.018

Table 3: Numerical values of the 4-class window composition.

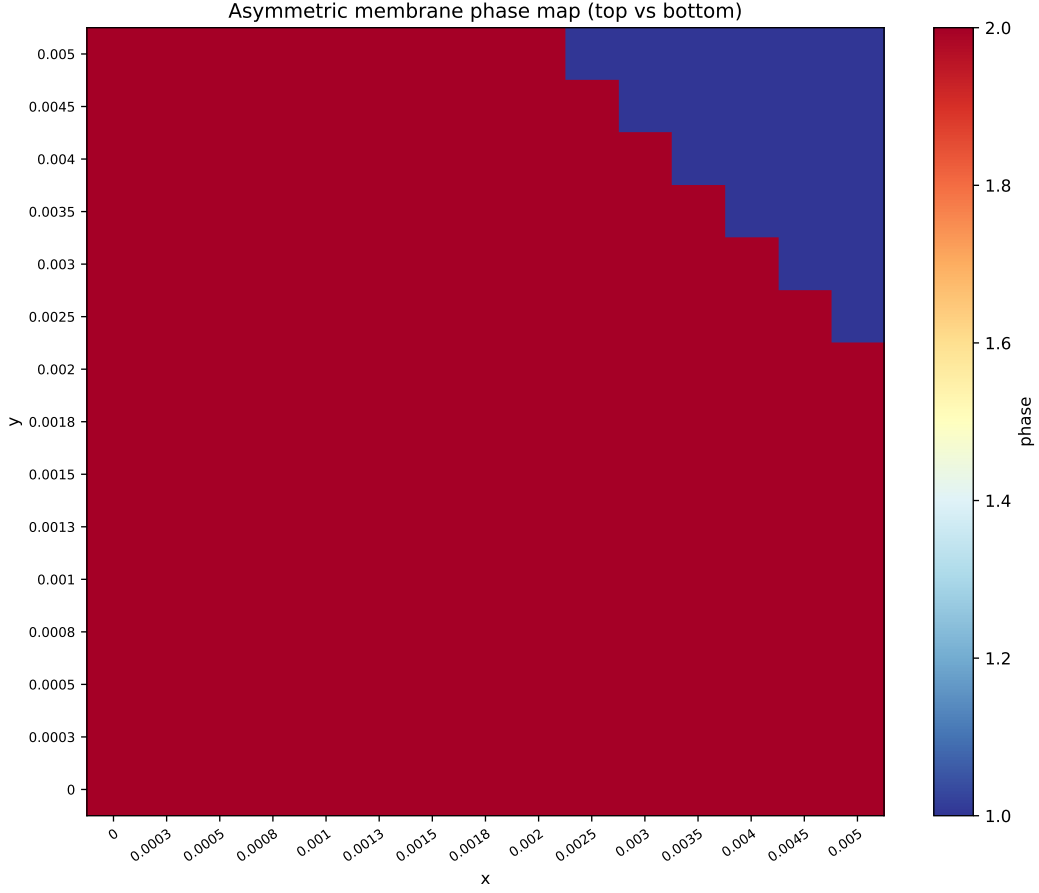


Figure 3: Asymmetric membrane phase map ($\kappa_{\uparrow}$ vs $\kappa_{\downarrow}$).

Case	q^*	Window	dominant class	w_{L1R1}
sym	0.4	peak1	L0R0	0.000
sym	0.4	valley	L0R0	0.197
sym	0.4	peak2	L0R1	0.104
asym	0.4	peak1	L0R0	0.000
asym	0.4	valley	L0R0	0.113
asym	0.4	peak2	L0R1	0.052
sym	0.5	peak1	L0R0	0.000
sym	0.5	valley	L0R0	0.004
sym	0.5	peak2	L0R1	0.036
asym	0.5	peak1	L0R0	0.000
asym	0.5	valley	L0R0	0.002
asym	0.5	peak2	L0R1	0.018
sym	0.6	peak1	L0R0	0.000
sym	0.6	valley	L0R0	0.002
sym	0.6	peak2	L0R1	0.022
asym	0.6	peak1	L0R0	0.000
asym	0.6	valley	L0R0	0.001
asym	0.6	peak2	L0R1	0.011

Table 4: q^* robustness check (exact decomposition): dominant class and w_{L1R1} across thresholds.

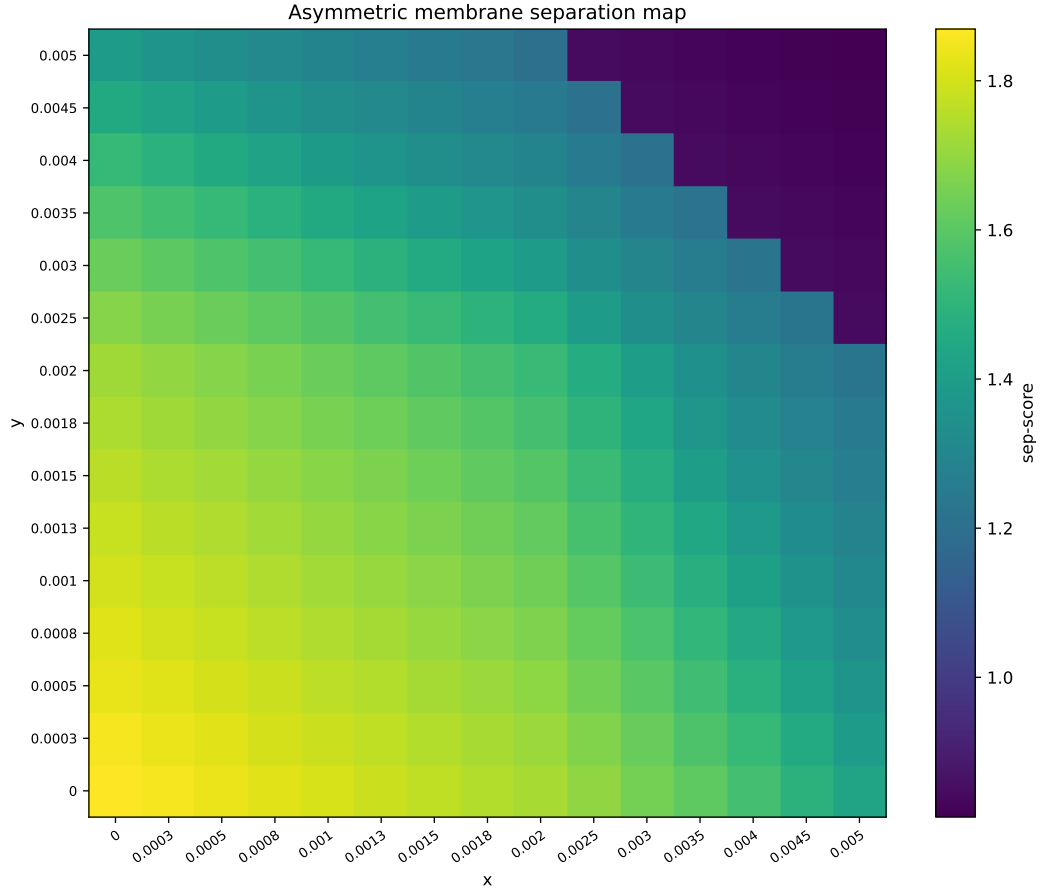


Figure 4: Asymmetric membrane separation map.

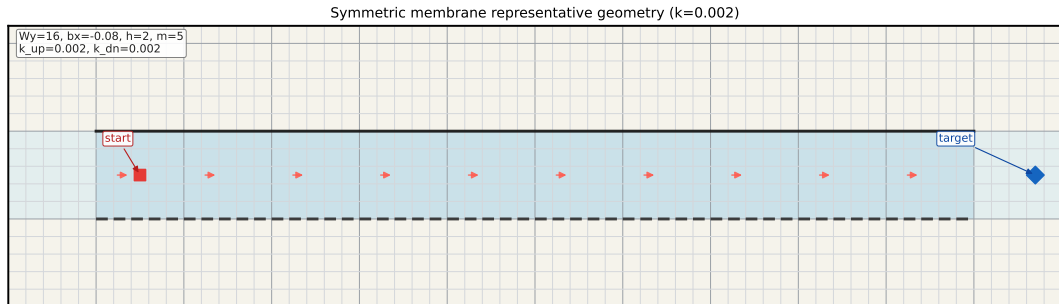


Figure 5: Symmetric membrane representative geometry (corridor band, membrane location, start/target, local-bias direction).

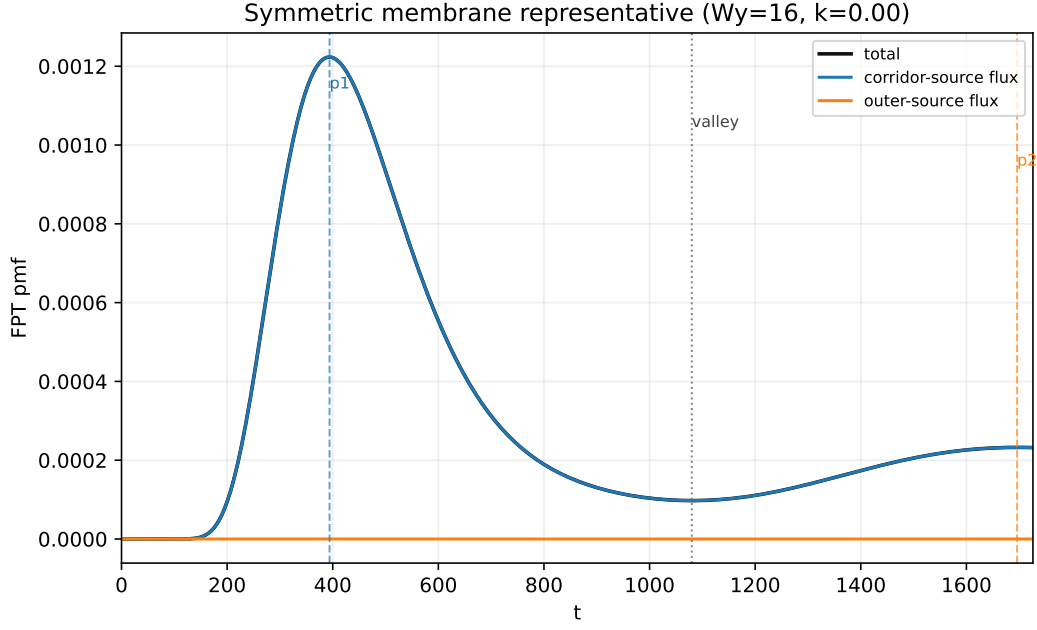


Figure 6: Symmetric representative ($\kappa = 0.002$): total FPT and corridor/outer-source flux decomposition.

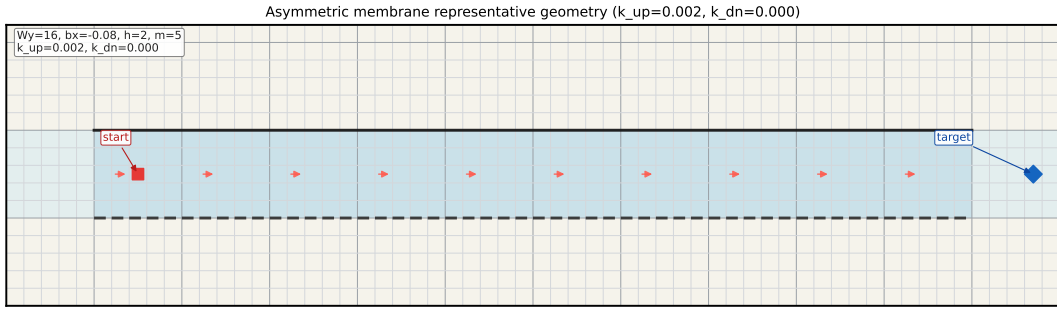


Figure 7: Asymmetric membrane representative geometry (different top/bottom membrane permeabilities).

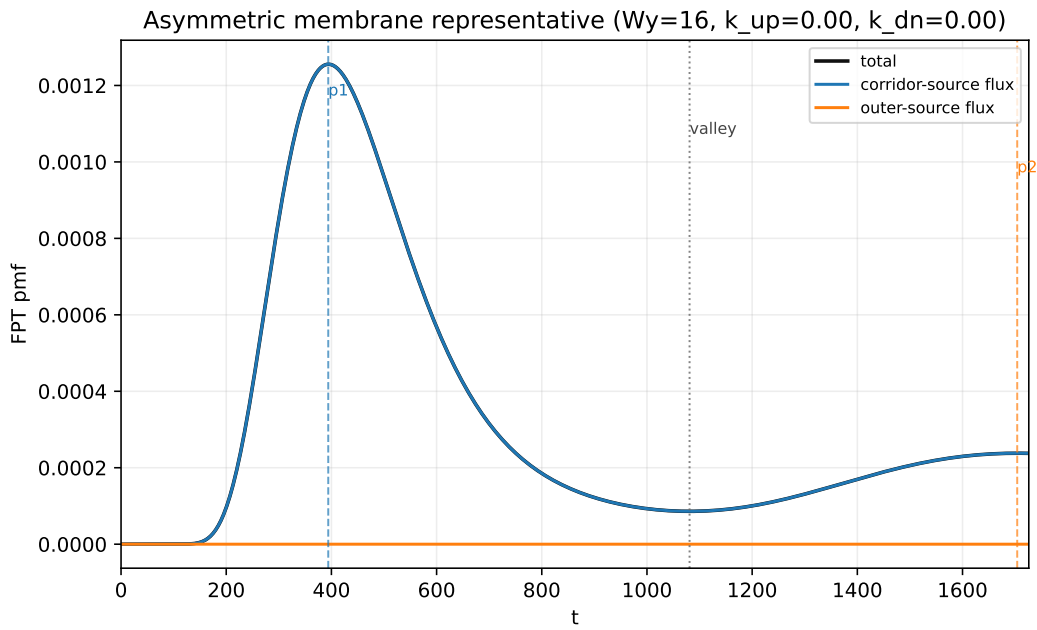


Figure 8: Asymmetric representative ($\kappa_{\uparrow} = 0.002, \kappa_{\downarrow} = 0$): total FPT and flux decomposition.

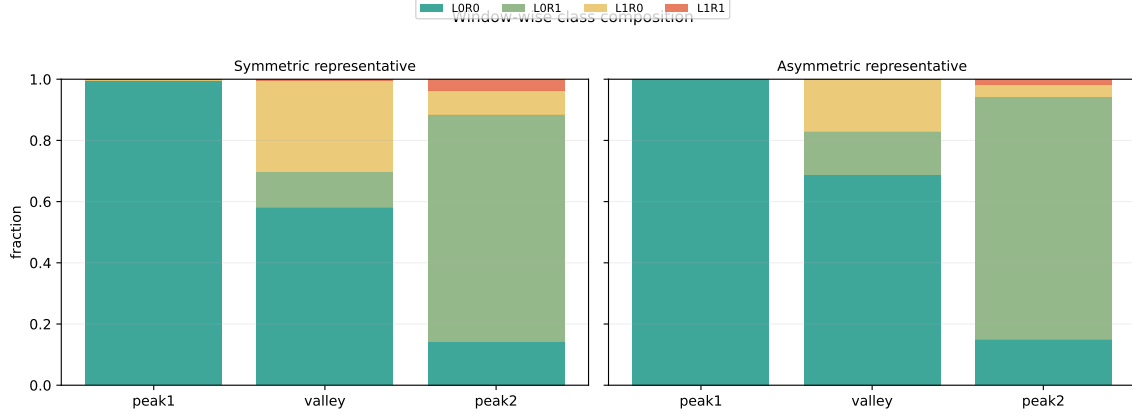


Figure 9: Window-wise composition of 4 classes ($L \times R$).

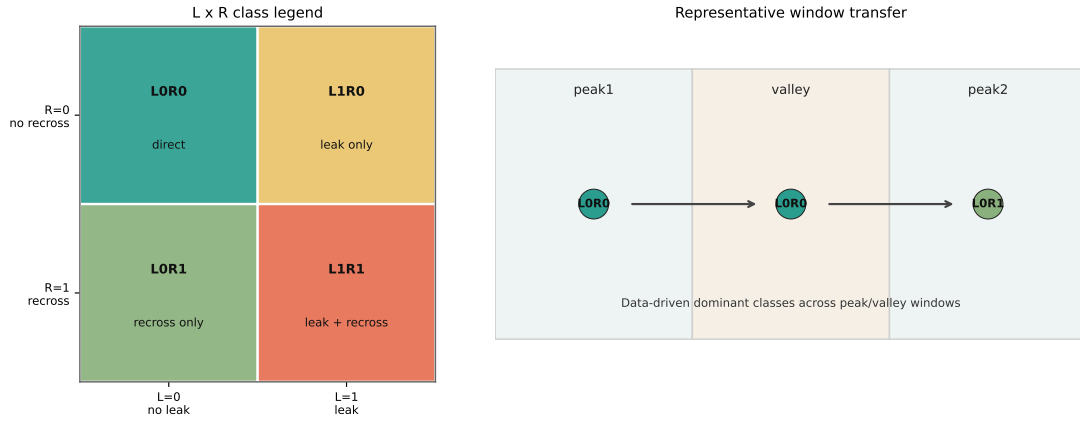


Figure 10: Representative $L \times R$ legend: left, 2x2 class definition; right, typical dominant-class transfer across peak1/valley/peak2 windows.

b_x	clear d interval	best d	phase(best)	$P(\text{near})$	$P(\text{far})$
+0.00	-	3	0	0.867	0.087
+0.02	-	2	0	0.603	0.383
+0.04	3-9	3	2	0.444	0.556
+0.05	2-10	2	2	0.393	0.607
+0.06	2-10	2	2	0.355	0.645
+0.07	2-10	2	2	0.326	0.674
+0.08	2-10	2	2	0.301	0.699
+0.09	2-10	2	2	0.281	0.719
+0.10	2-10	2	2	0.264	0.736
+0.11	2-10	2	2	0.248	0.752
+0.12	2-10	2	2	0.235	0.765
+0.14	2-10	2	2	0.212	0.788
+0.16	2-10	3	2	0.206	0.794

Table 5: Per- b_x summary: clear intervals and best near-target distance.

spec	b_x	d	Δy	phase	sep-score	valley/max	$p_{\min}$	peak-ratio	peak-margin	min-margin	gate-pass	gate-slack	clear-score	showcase	ϵ_{split}
adaptive_refine	+0.12	4	1	2	2.38	0.007	0.350	0.170	0.070	0.070	yes	0.070	1.54	1.92	1.15×10^{-9}

Table 6: No-corridor clear-instance quality metrics (auto-selected from all candidate scans). We report **peak-margin**, **min-margin**, and strong-gate audit fields (**gate-pass/gate-slack**) to avoid near-threshold pseudo-clear picks, and include **showcase** for visual ranking consistency.

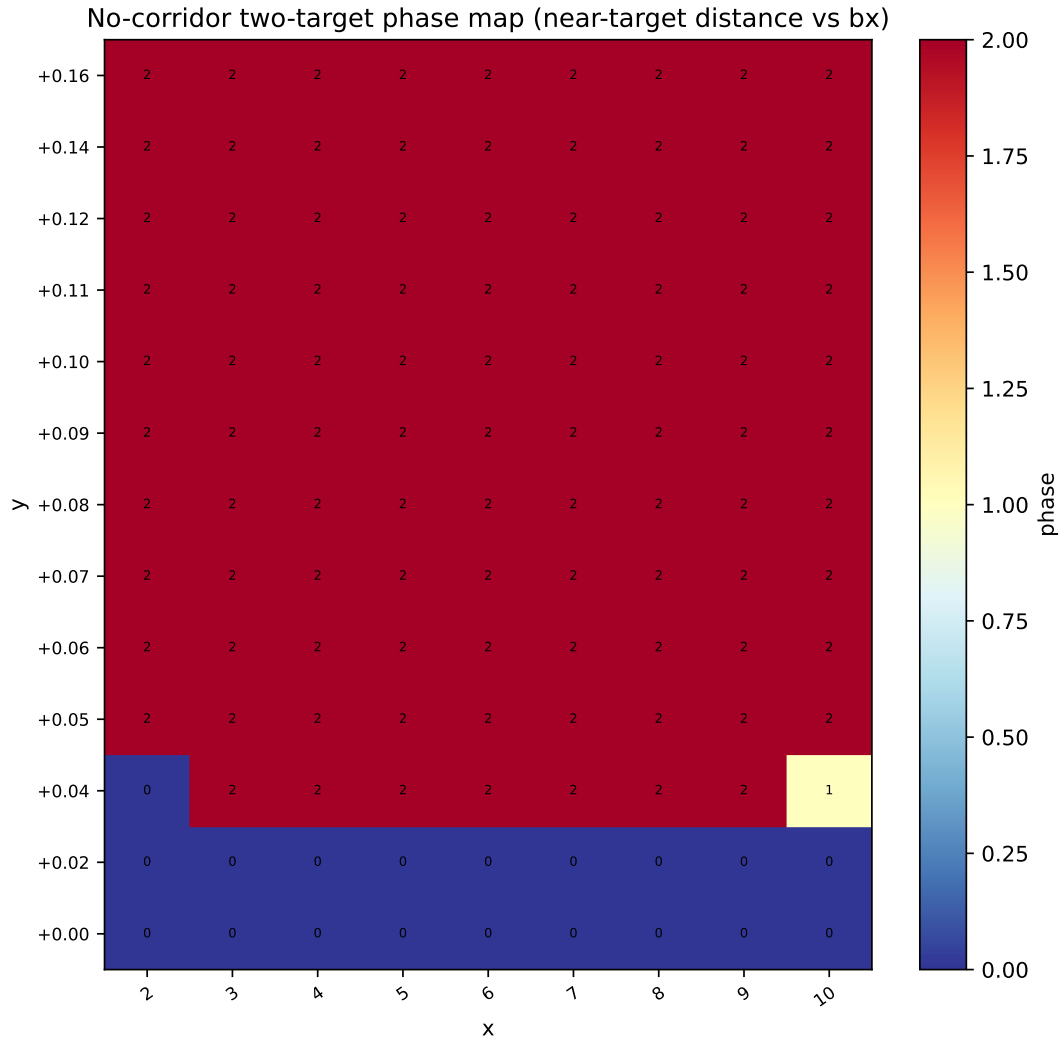


Figure 11: No-corridor two-target phase map (near-target distance d vs b_x).

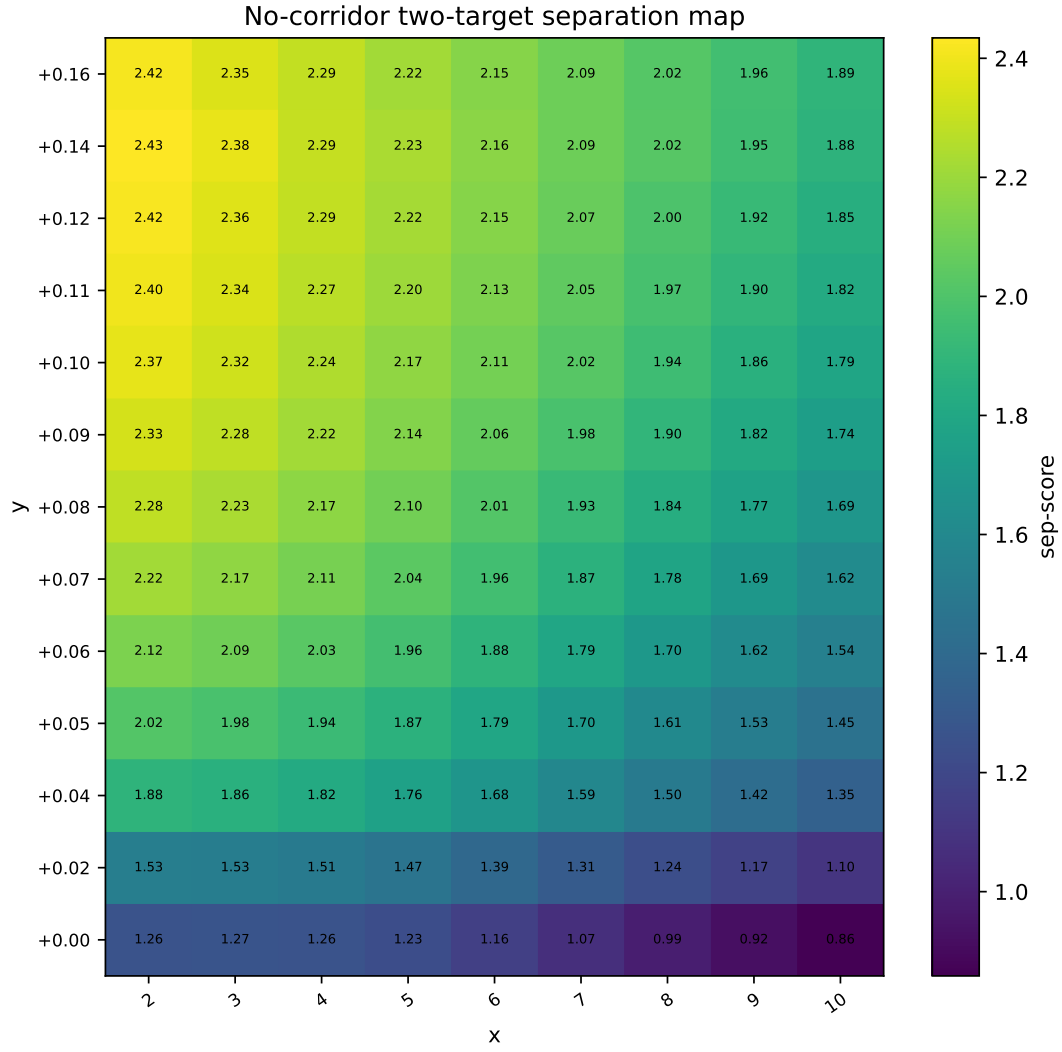


Figure 12: No-corridor two-target separation map.

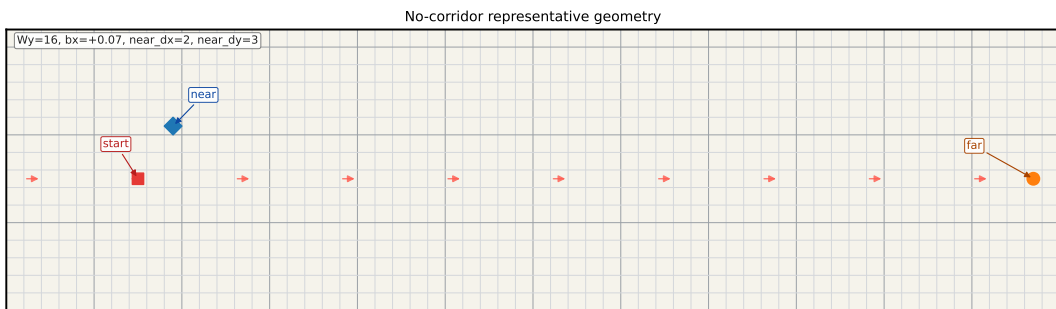


Figure 13: Representative no-corridor geometry (start, near target, far target, and global drift direction).

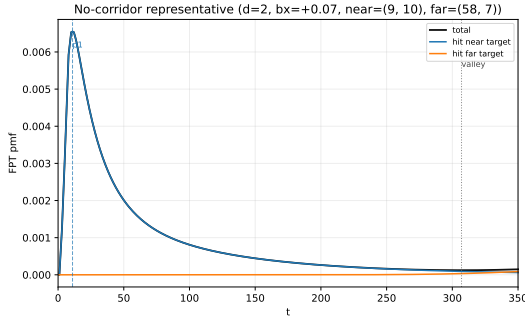


Figure 14: Representative no-corridor case: total FPT with near/far splitting components.

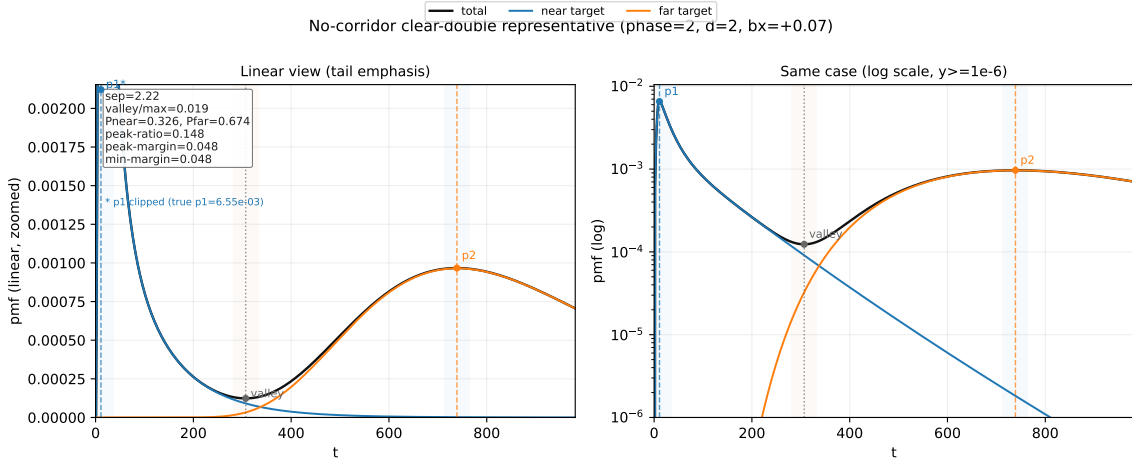


Figure 15: No-corridor clear-double exemplar: left, a tail-emphasized linear view (peak1 is clipped and marked as p1* for readability of valley/peak2); right, a log view with $y \geq 10^{-6}$.

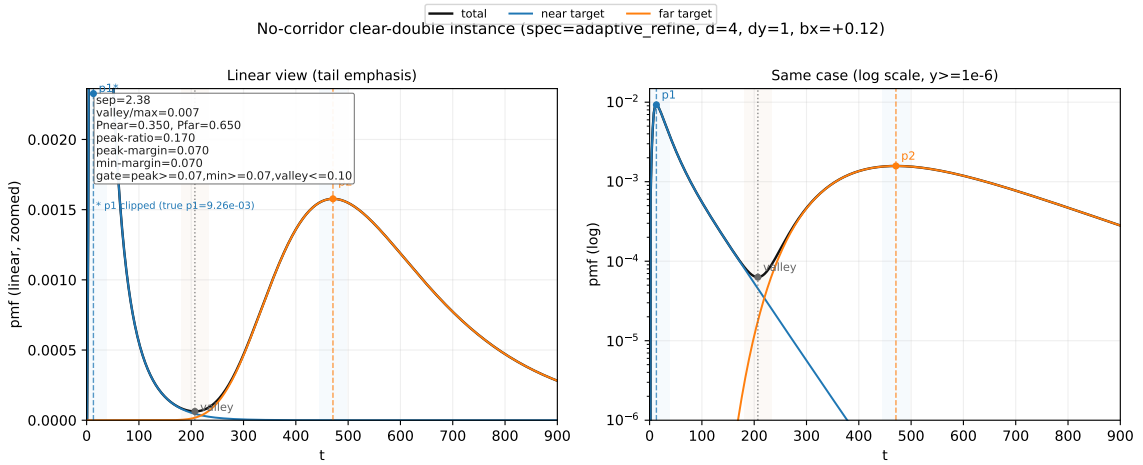


Figure 16: Dedicated no-corridor clear-instance figure (maximum showcase-score over all candidate scans): tail-emphasized linear panel plus log panel to keep valley/peak2 structure explicit.

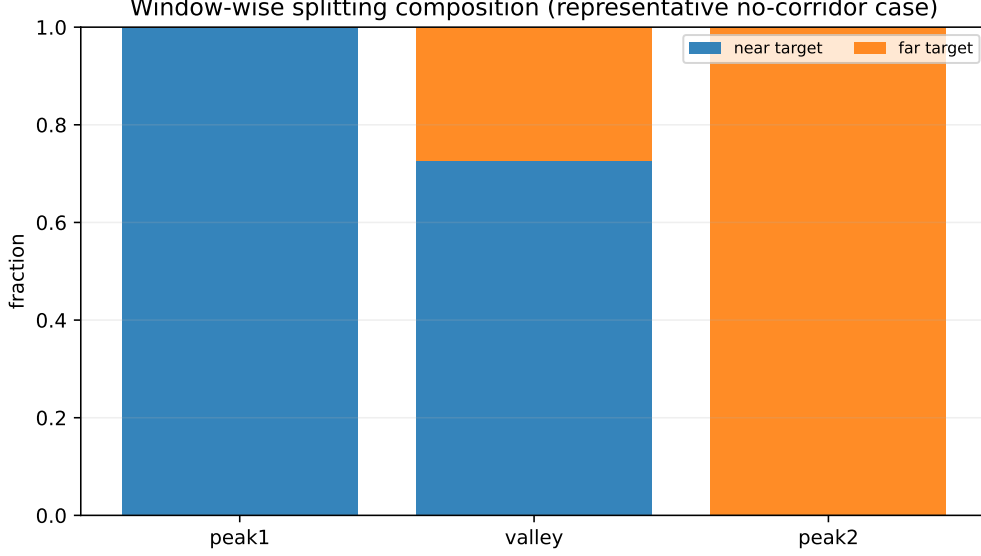


Figure 17: Window-wise near/far mass fractions in the representative case.

comes from the adaptive-refinement round, $(b_x, d, \Delta y) = (0.12, 4, 1)$, with `peak-margin`=0.070, `min-margin`=0.070, and `showcase`=1.92. It passes the first gate:

$$\text{peak-margin} \geq 0.07, \quad \text{min-margin} \geq 0.07, \quad \text{valley/max} \leq 0.10.$$

Its primary-gate audit reports `gate-pass=yes`, `gate-slack`=0.070 (minimum slack over the three thresholds), and 168 candidates in the first gate pool.

4 Method Reference (Committor + 2x2 Classes)

Reproducible Definition of Phase and sep-score

To make phase maps fully reproducible, we use the same criteria as in `src/vkcore/grid2d/rect_bimodality/cli.py`:

(1) Two-peak detection (`find_two_peaks`) Let $\tilde{f} = \text{smooth}_7(f)$ (window-7 smoothing). Candidate peaks satisfy

$$\tilde{f}(t) > \tilde{f}(t-1), \quad \tilde{f}(t) \geq \tilde{f}(t+1), \quad \tilde{f}(t) \geq 0.06 \max \tilde{f}.$$

Pair selection then enforces (default code thresholds)

$$t_{p2} - t_{p1} \geq 20, \quad \text{peak_balance} \geq 0.08, \quad \text{valley_drop} \geq 0.08, \quad \text{minor_drop} \geq 0.005,$$

plus side-dominance tolerance (`dominance_tol`=0.02). Define

$$\text{has_double} := \mathbf{1}\{t_{p1}, t_{p2} \text{ both exist}\}.$$

(2) One-target phase definition Define

$$\text{sep}_{\text{peaks}} := \frac{|t_{p2} - t_{p1}|}{w_{1/2}^{(1)} + w_{1/2}^{(2)}}, \quad \text{valley/max} := \frac{h_v}{\max(h_{p1}, h_{p2})}, \quad \text{peak_balance} := \frac{\min(h_{p1}, h_{p2})}{\max(h_{p1}, h_{p2})}.$$

Here $w_{1/2}^{(k)}$ is half-width at half-maximum of peak k (computed on raw f), while h_{p1}, h_{p2}, h_v are read from smoothed $\tilde{f}$. Then

$$\text{phase}_{\text{OT}} = \begin{cases} 2, & \text{has_double} = 1, \text{sep}_{\text{peaks}} \geq 1.0, \text{valley/max} \leq 0.35, \text{peak_balance} \geq 0.15, \\ 1, & \text{has_double} = 1 \text{ but the above clear-double test fails,} \\ 0, & \text{has_double} = 0. \end{cases}$$

(3) **Two-target phase definition** Define modal times and mode-separation score

$$t_{\text{mode}}^{\text{near}} := \arg \max_t f_{\text{near}}(t), \quad t_{\text{mode}}^{\text{far}} := \arg \max_t f_{\text{far}}(t),$$

$$\text{sep}_{\text{mode}} := \frac{|t_{\text{mode}}^{\text{far}} - t_{\text{mode}}^{\text{near}}|}{w_{1/2}^{\text{far}} + w_{1/2}^{\text{near}}}.$$

Define normalized splitting masses

$$P_{\text{near}} := \frac{\sum_t f_{\text{near}}(t)}{\sum_t f_{\text{any}}(t)}, \quad P_{\text{far}} := \frac{\sum_t f_{\text{far}}(t)}{\sum_t f_{\text{any}}(t)}.$$

On the total law, define

$$\text{peak_ratio} := \frac{\min(h_{p1}, h_{p2})}{\max(h_{p1}, h_{p2})}, \quad \text{valley}/\text{max} := \frac{h_v}{\max(h_{p1}, h_{p2})}.$$

Then

$$\text{phase}_{\text{TT}} = \begin{cases} 2, & \text{has_double} = 1, \text{sep}_{\text{mode}} \geq 1.0, \min(P_{\text{near}}, P_{\text{far}}) \geq 0.15, \\ & \text{valley}/\text{max} \leq 0.35, \text{peak_ratio} \geq 0.10, (\text{peak_ratio} - \text{valley}/\text{max}) \geq 0.01, \\ 1, & \text{has_double} = 1 \text{ but the above clear-double test fails,} \\ 0, & \text{has_double} = 0. \end{cases}$$

So mathematically, **phase=1** means “double peaks are detected, but at least one clear-double threshold is not met”.

Committer Mechanism (Detailed)

Let the hitting time be $\tau_C := \inf\{t \geq 0 : X_t \in C\}$. On the discrete Markov chain with transition matrix P , let A be the start basin (here a strip near the start) and B be the absorbing target set. Define

$$q_i := \mathbb{P}_i(\tau_B < \tau_A),$$

which solves the discrete Dirichlet boundary-value problem

$$q_i = 0 \ (i \in A), \quad q_i = 1 \ (i \in B), \quad q_i = \sum_j P_{ij} q_j \ (i \notin A \cup B).$$

With $U = \Omega \setminus (A \cup B)$, this is equivalent to the linear system

$$(I - P_{UU}) q_U = P_{UB} \mathbf{1}.$$

For this absorbing setup the solution is unique, so q is a dynamics-determined progress coordinate (not an ad hoc geometric proxy). We use q as the progress coordinate and set $q^* = 0.5$ to define the commitment surface

$$\Sigma_{q^*} := \{i : q_i \geq q^*\}.$$

Moreover, when $\rho(P_{UU}) < 1$, the Neumann expansion gives

$$q_U = (I - P_{UU})^{-1} P_{UB} \mathbf{1} = \sum_{n \geq 0} P_{UU}^n P_{UB} \mathbf{1}.$$

So q_U is the cumulative contribution of paths that spend n interior steps before absorption into B , which provides a direct linear-system bridge to the renewal failure-retry interpretation used later: early peak mass is small- n dominated, while the slow peak accumulates from larger- n loops.

L×R Events and Recross Counts

For a trajectory X_t up to τ_B , define membrane-cross count

$$N_{\text{mem}} := \sum_{t=0}^{\tau_B-1} \mathbf{1}\{(X_t, X_{t+1}) \in \mathcal{E}_{\text{mem}}\},$$

and commitment-to-start recross count

$$N_{\text{ret}} := \sum_{t=0}^{\tau_B-1} \mathbf{1}\{X_t \in \Sigma_{q^*}, X_{t+1} \in A\}.$$

Then use binary labels

$$L = \mathbf{1}\{N_{\text{mem}} \geq 1\}, \quad R = \mathbf{1}\{N_{\text{ret}} \geq 1\},$$

which yields four classes

$$(L, R) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\},$$

implemented as LOR0/LOR1/L1R0/L1R1 (Figure 10).

Peak/Valley Mechanism as Renewal Decomposition

Let $f(t)$ be the FPT law. With direct-success kernel $g(t)$ and fail-return kernel $h(t)$,

$$f(t) = g(t) + (h * f)(t) = \sum_{k \geq 0} (h^{*k} * g)(t).$$

Here k is the number of failed attempts before success. Bimodality appears when the $k = 0$ (near one-shot) and $k \geq 1$ parts separate in both timescale and mass. Window fractions

$$w_{\ell r}(W) = \frac{\sum_{t \in W} f_{\ell r}(t)}{\sum_{t \in W} f(t)}$$

make the transfer explicit: peak1 is mainly LOR0; valley remains LOR0-major with a clear rise of L1R0; and peak2 is LOR1-dominant, with L1R1 secondary (Figure 9, Table 3).

Closed-Form Bridge from Linear System to FPT Kernels

For the no-corridor two-target branch, let U denote interior transient states, e_s the interior basis vector at the start, and

$$Q := P_{UU}, \quad R_{\text{near}} := P_{U, \partial_{\text{near}}}, \quad R_{\text{far}} := P_{U, \partial_{\text{far}}}.$$

Then the first-hit pmfs are

$$f_{\text{near}}(t) = e_s^\top Q^{t-1} R_{\text{near}} \mathbf{1}, \quad f_{\text{far}}(t) = e_s^\top Q^{t-1} R_{\text{far}} \mathbf{1}, \quad t \geq 1.$$

Hence the far-hit mass is

$$P_{\text{far}} = \sum_{t \geq 1} f_{\text{far}}(t) = e_s^\top (I - Q)^{-1} R_{\text{far}} \mathbf{1}.$$

This gives the exact bridge between time-domain decomposition and the linear-system view used by committors.

Conditioned Kernel (Doob Transform) and Peak Transfer

On interior states with $q_{\text{far}}(i) > 0$, define

$$\tilde{P}_{ij}^{\text{far}} := \frac{P_{ij} q_{\text{far}}(j)}{q_{\text{far}}(i)}.$$

This is the Doob h -transform of P by $h = q_{\text{far}}$, i.e., the exact dynamics conditioned on eventually hitting *far* before *near*. In this view, the near branch is the complementary conditioned dynamics through $q_{\text{near}} = 1 - q_{\text{far}}$. Therefore, the window transfer from peak1 to peak2 can be interpreted as a measure transfer between two conditioned ensembles: early-time mass is dominated by short successful excursions, while late-time mass is amplified by loops that repeatedly leave and re-enter low- q_{far} regions before final absorption. This provides a direct dynamical meaning for the observed renewal split.

Spectral View: Origin of Fast/Slow Timescales

On the interior kernel Q , write the (generalized) spectral expansion

$$Q^{t-1} = \sum_k \lambda_k^{t-1} r_k \ell_k^\top, \quad |\lambda_1| > |\lambda_2| \geq \dots$$

Then

$$f_{\text{near}}(t) = \sum_k c_k^{\text{near}} \lambda_k^{t-1}, \quad f_{\text{far}}(t) = \sum_k c_k^{\text{far}} \lambda_k^{t-1}.$$

Modes with smaller $|\lambda_k|$ decay quickly and dominate early windows, while the leading mode λ_1 controls the slow tail and late peak. If near/far branches project differently on these modes, we naturally obtain “near-dominant early window” versus “far-dominant late window” transfer. This is fully consistent with the renewal picture: peak1 is short-stay/low-loop mass, while peak2 is quasi-stationary release from the leading mode. Equivalently, the conditioned hazard

$$h_{\text{far}}(t) := \frac{f_{\text{far}}(t)}{S(t-1)}$$

increases over time when the far channel progressively takes over late windows.

Splitting Committor Closure for Two Targets

On the same system, define

$$q_{\text{far}}(i) := \mathbb{P}_i(\tau_{\text{far}} < \tau_{\text{near}}), \quad q_{\text{near}}(i) = 1 - q_{\text{far}}(i).$$

Its boundary-value problem is

$$q_{\text{far}} = 0 \text{ on near}, \quad q_{\text{far}} = 1 \text{ on far}, \quad q_{\text{far}}(i) = \sum_j P_{ij} q_{\text{far}}(j) \text{ on interior.}$$

This gives a direct closure with the exact decomposition $f = f_{\text{near}} + f_{\text{far}}$: $q_{\text{far}}(x_s)$ should match P_{far} . Define

$$\varepsilon_{\text{split}} := \left| q_{\text{far}}(x_s) - \sum_{t \geq 1} f_{\text{far}}(t) \right| = |q_{\text{far}}(x_s) - P_{\text{far}}|.$$

Moreover, let $\tau := \tau_{\text{near}} \wedge \tau_{\text{far}}$ and define

$$M_t := q_{\text{far}}(X_{t \wedge \tau}).$$

Because q_{far} is harmonic on interior states and fixed to 0/1 on near/far boundaries, M_t is a bounded martingale. Optional stopping then gives

$$q_{\text{far}}(x_s) = \mathbb{E}[M_\tau] = \mathbb{P}_{x_s}(\tau_{\text{far}} < \tau_{\text{near}}) = P_{\text{far}}.$$

So $\varepsilon_{\text{split}}$ is theoretically zero; numerically, its scale should track the Bellman-residual scale, making it a joint diagnostic for mechanism closure and solver stability.

case	$q_{\text{far}}(x_s)$	P_{far}	$\varepsilon_{\text{split}}$	$\ r_{\text{int}}\ _\infty$	$\ r_{\text{near}}\ _\infty$	$\ r_{\text{far}}\ _\infty$
representative	0.674289	0.674289	1.97×10^{-8}	9.93×10^{-12}	0	0
clear-instance	0.650264	0.650264	1.15×10^{-9}	9.93×10^{-12}	0	0

Table 7: Splitting-committor closure QC for no-corridor two-target runs (representative + dedicated clear-instance): we report $q_{\text{far}}(x_s)$, P_{far} , $\varepsilon_{\text{split}}$, and interior/boundary residual norms.

Table 7 shows that both routes satisfy $q_{\text{far}}(x_s) \approx P_{\text{far}}$, with $\varepsilon_{\text{split}}$ at 10^{-8} and 10^{-9} scales, while interior Bellman residuals stay at the 10^{-11} scale. This closes the mechanism loop behind the fast/slow peak split for both representative and showcase-quality clear-instance routes. We track these residuals together with $\varepsilon_{\text{split}}$ as automatic QC scalars in continuous optimization runs.

Committor Implementation in This Report

1. Build the exact transition kernel for each representative case (including laziness, drift, membrane permeability, and reflecting boundaries).
2. Use $A = \{x \leq x_s + 1, |y - y_{\text{mid}}| \leq 1\}$ and $B = \{\text{target}\}$, and solve q via fixed-point iteration with tolerance 10^{-11} ; record interior Bellman residual and boundary mismatch as quality diagnostics.
3. Use exact augmented-state recursion on (x, ℓ, s, r) (not MC): ℓ tracks leak, s tracks whether Σ_{q^*} has been visited, and r tracks strict recross (return to A after visiting Σ_{q^*}).
4. Report $q^* = 0.5$ as default and additionally provide $q^* \in \{0.4, 0.5, 0.6\}$ robustness results (Table 4).
5. For no-corridor two-target cases, keep the strict identity $f = f_{\text{near}} + f_{\text{far}}$ as level-1 decomposition, then interpret windows with committor-consistent labels.

5 Takeaways and Next Steps (Committor-Centric)

Implementation Summary

1. Membrane corridor track: clear-double persists in both symmetric and asymmetric representatives; under strict recross, peak1 is mostly LOR0, valley gains L1R0, and peak2 is mostly LOR1.
2. No-corridor two-target track: we now run “all-candidates scan $\rightarrow$ adaptive local refinement $\rightarrow$ robust-gated clear-instance selection” and report a dedicated figure/table pair (Figure 16, Table 6); the selected clear-instance is $(b_x, d, \Delta y) = (0.12, 4, 1)$, keeps positive phase=2 margins, and passes the first robustness gate (peak-margin ≥ 0.07 , min-margin ≥ 0.07 , valley/max ≤ 0.10). We additionally export strong-gate audit scalars (**gate-pass/gate-slack**) plus first-gate candidate count (168) and keep the hard clear-figure integrity check (file exists and size $> 2\text{KB}$), with figure path/size metadata written to ‘summary.json’ for continuous monitoring.
3. Methodologically, committor is now fully operational in the pipeline; splitting committor is numerically closed against exact far-hit mass for both representative and clear-instance

cases, and we jointly monitor $\varepsilon_{\text{split}}$ plus committor residual diagnostics in automatic QC (current $\varepsilon_{\text{split}}$ scale $\sim 10^{-8}$ for representative and $\sim 10^{-9}$ for clear-instance).

4. Operationally, the current full local regression (codegen + CN/EN build checks) takes about 70s per round (around 1.2 minutes), so it stays in the local-first regime (< 10 minutes); we switch to Isambard only when projected single-round runtime exceeds 20 minutes.

Next Directions (All Grounded in Committor)

1. Scale exact decomposition from representative cases to full scan maps, and export threshold-margin maps for clear-instance quality (not just single-point tables).
2. Run q^* sensitivity analysis (e.g., 0.4/0.5/0.6) and report peak/valley class stability.
3. Systematically output splitting committor fields and window summaries for two-target scans:

$$q_{\text{far}}(x) = \mathbb{P}_x(\tau_{\text{far}} < \tau_{\text{near}}),$$

and use both $\varepsilon_{\text{split}} = |q_{\text{far}}(x_s) - P_{\text{far}}|$ (for representative and clear-instance routes) and **min-margin/gate-slack** as automatic quality checks.

4. Compare top/bottom membrane asymmetry versus directional asymmetry (inside/outside permeability mismatch) in a unified committor geometry view.
5. Keep continuous build checks active (codegen + EN/CN LaTeX) and enforce routing policy in keepalive: run local for short rounds (< 10 minutes), and use isbard doctor/auth + submit/status/fetch for large scans (projected round > 20 minutes), with automatic local-smoke fallback on remote failures.
6. For every remote round, always record JOB_ID, REMOTE_DIR, key artifact paths, and explicit next actions in the run summary.

Reproducibility

```
cd reports/grid2d_membrane_near_target
../../.venv/bin/python code/membrane_near_target_report.py
latexmk -pdf -interaction=nonstopmode -halt-on-error \
  -auxdir=build -emulate-aux-dir grid2d_membrane_near_target_en.tex
```