Online Supplement

When the Benchmark Moves with the Policy: Measuring Access in Allocation Simulations

Pablo Loschi · Version 7.12 · 19 November 2026

This Online Supplement preserves all extended methodological documentation, proofs, stress-test tables, figures, New York City proxy scenarios, reproducibility details, and the quarantined PolicySpace2 investigation accompanying the main Technical Note.

Supplement S1. Excluded PolicySpace2 Extension (Quarantined)

This section preserves an unverified archived analysis and its failed-replay record strictly for transparency. None of its numerical comparisons has been reproduced from the original monthly histories, and none supports the abstract, main evidential claims, or conclusions of the Technical Note.

S1.1 Archived comparison

PolicySpace2 (Furtado 2022) is a published agent-based model of a Brazilian metropolitan economy with open code. Furtado compares three poverty policies funded by the same share of municipal revenue: buying houses and giving them to the poorest registered families, 24-month rental vouchers, and a monthly cash transfer shared among all registered families. The paper reports that "on average, Monetary aid policy helps 2,358.05 households monthly per municipality, compared to 47.5 for Rental vouchers and 12.37 for the Property acquisition program" (Furtado 2022, para. 5.3), describing monetary aid as reaching "a larger number of households" (para. 5.9). Because that headcount resets each month, monetary aid counts everyone on each monthly register whereas property acquisition counts only the houses handed over during that month.

The archived audit used the published code at its default Brasília setting, adding instrumentation intended to log eligibility, assistance, and poverty status without altering policy behaviour (Section S1.2 documents provenance, missing logs, and non-determinism). Saved summaries sit close to the paper's monetary-aid figure at 2,363 households a month on average (2,353–2,373 across replications), alongside 54 for vouchers and 14 for property acquisition.

Table S1.1 recounts those archived runs along three distinct denominators. On the monthly flow count, monetary aid reaches about 170 times as many households as property acquisition in the archived summaries (190 times in Furtado's published figures), whereas counting distinct families assisted at least once over the 108 active-policy months (2011–2019) reduces the ratio to 2.3 because monetary aid repeatedly pays the same families (75 payments per family on average) while each house transfers to a new family. A third cohort metric tracks families first falling below the eligibility line in months 13 to 36 and checks how many remain below that threshold 24 months later: without intervention 93.6% remain below, compared with reductions of 1.8 points (SE 0.7) for monetary aid, 2.6 (0.4) for rental vouchers, and 6.9 (0.7) for property acquisition across paired seeds, describing different outcomes across different denominators in the archived summaries.

Policy	Households helped per month (reported)	Distinct families helped, 2011–2019	Newly poor helped within 24 months	Newly poor still below the line after 24 months
No policy	–	–	–	93.6% [91.5, 94.8]
Monetary aid	2,363 [2,353, 2,373]	3,405 [3,354, 3,479]	100%	91.8% [88.5, 93.9]
Rental vouchers	54 [52, 56]	1,712 [1,659, 1,786]	57.5% [55.8, 62.7]	91.0% [89.1, 94.5]
Property acquisition	14 [12, 15]	1,468 [1,320, 1,586]	17.0% [13.5, 19.9]	86.7% [84.3, 88.1]

Table S1.1. PolicySpace2 policies counted three ways (unverified archived summaries for Brasília, 1% of the population, 2010–2020 at default parameters with policies starting after the first year, reporting means over ten replications and ranges across replications in brackets). Newly poor refers to families whose permanent income first falls below the 20th percentile in months 13–36 (395 per run without policy and 673 under property acquisition).

These PolicySpace2 comparisons remain provisional because the archived simulations have not been reproduced, and the entry-cohort contrast requires the same population caution emphasised in Section 4 of the main paper. Property acquisition alters who falls below the poverty line, raising its entry cohort to 673 families per run against about 400 under the other policies. Its lower persistence therefore partly reflects compositional shifts instead of pure exit effects, while across ten replications voucher and monetary-aid persistence cannot be distinguished statistically (paired difference 0.8 points, SE 0.7). The 2.3 ratio and apparent reversal are unverified properties of archived summaries that cannot serve as evidence about real-world policies without the original monthly histories or a matching replay.

S1.2 Provenance and structural non-determinism

The original audit uses commit `ced77dd`, Brasília, 1% population, 2010–2020, policy share $\delta = 0.2$, and eligibility below the 20th percentile of permanent income, with seeds 1–10 for each policy and a compatibility shim targeting Python 3.6 `random.sample` behaviour on dictionary views. While the saved per-run summaries reconstruct Table S1.1, exhaustive checks across local directories, Docker containers and volumes, Git history, and all four benchmark-audit Zenodo versions found none of the 40 original monthly logs (`audit_*.jsonl`); the lone saved `audit_wage_1.jsonl` (`12,476,844` bytes) in `Methods_v7_3_audit_evidence.zip` matches the later 11 November local Docker replay byte-for-byte.

Two full 120-month replays of monetary aid (`wage`, seed `1`) in isolated Python 3.6 environments with identical 186-package Linux conda stacks illustrate why Table S1.1 cannot be reproduced historically. In the 11 November Docker replay, eight of the eleven archived summary fields differed (mean monthly assistance `2360.9` vs. `2357.9`, total assistance counts `254,981` vs. `254,649`, entry cohort `409` vs. `382`), halting any multi-seed extension; that run is retained under `policyspace2/later_replay/` in the audit-evidence archive. On 15 November, a second independent rebuild from the same package manifest (`PYTHONHASHSEED` unset, `1,403s`) produced a third distinct realisation at `256,334` total assistance counts (`2,373.5` monthly, `444` cohort, `0.8986` still below the threshold after 24 months). Neither later run replaces the archived Table S1.1 summaries.

The repaired analyser, `analyze_ps2_audit.py`, enforces the full declared policy-by-seed file set, consecutive calendar months, and complete follow-up instead of returning an empty success on missing logs, includes zero-assistance months in the calendar denominator, and counts any assistance from cohort entry through observation 24 months later. Although that timing rule guards against histories recording assistance prior to the first logged poverty month, synthetic regression cases confirm its effect only in principle because no such case occurs in the retained replay and its incidence in the original runs cannot be verified without their logs. On the saved replay run, the repaired analyser leaves all eleven shared summary fields unchanged and all ten unit-test cases in `test_audit_contract.py` pass.

Inspection of Docker metadata and source files alongside seven controlled container probes (`SEED=1`, `POLICIES="wage"`, same `186`-package image) identifies three structural mechanisms behind the divergence inside unmodified `ced77dd`. First, Python hash randomisation (`PYTHONHASHSEED`) alters unordered `set` iteration orders across `simulation.py` (`mun_to_regions`), `agents/family.py` / `world/demographics.py` (`relatives`), and `world/funds.py` (`set(Family)` sorting): loading identical cached `.agents` at `SEED=1` and changing only `PYTHONHASHSEED` from `1` to `2` immediately shifts initial vacant and total houses (`909 / 11,267` vs. `915 / 11,266`) and Months 1–3 poverty headcounts (`[2067, 2084, 2094]` vs. `[2059, 2081, 2092]`). Second, loading an existing `.agents` cache skips `create_regions()` and `create_all()` while keeping the same `self.seed` stream (`simulation.py`), immediately shifting initial states even with identical `PYTHONHASHSEED=1` and `SEED=1` (`909 / 11,267` cached vs. `913 / 11,265` fresh `run_a`), and the cache filename omits `SEED`. Third, `world/generator.py` draws `uuid.uuid4()[:12]` identifiers outside `self.seed`: across two paired 26-month probes (`2010-01-01` through `2012-02-01`, 14 active policy months) pinning both `.agents` and `PYTHONHASHSEED=1`, all five CSV outputs (`temp_banks.csv`, `temp_construction.csv`, `temp_firms.csv`, `temp_regional.csv`, `temp_stats.csv`) match byte-for-byte and all 26 monthly policy counts agree (`rep_sum = 36,726`), yet family IDs in `audit_wage_1.jsonl` still diverge on all 26 monthly observations (`diff=132` with `only_a=66, only_b=66` on `2010-01-01`, rising by `2012-02-01` to `1,706` for poverty registers and `1,818` for eligible and helped registers) as new marriages receive fresh random UUIDs. Exact family-level reproduction of Table S1.1 is therefore blocked structurally in unmodified `ced77dd` rather than by an unrecovered package lock, leaving PolicySpace2 strictly as a quarantined, non-evidential supplement unless upstream code is modified with deterministic identifiers, sorted set iterations, and seed-keyed caches for fresh multi-seed runs.

Supplement S2. Extended LivSim Sensitivity and Provenance

S2.1 LivSim audit provenance

The original three paired replications use the shipped synthetic inputs of LivSim (Kilambi, Bui and Mehrotra 2018), three simulated years, and NumPy seed 7777 separately for each policy and replication. Only input paths, the one-element NumPy acceptance comparison, and per-patient audit logging are patched, while the earlier extension completed the remaining two pairs. The five-pair summary and saved annual area counts used for the diagnostics are retained under `evidence/` in the accompanying replication package, and the audit-evidence archive provides all five pairs under `livsim/runs/` alongside instrumented source, input hashes, and run-time records. Only completed saved outputs inform these diagnostics; no additional LivSim simulation was run for this revision.

S2.2 Replication dispersion and leave-one-out sensitivity

The standard-library fast reconstruction is `python reproduce_review.py` (while optional `livsim_uncertainty.py` and `livsim_fixed_denominators.py` require SciPy and NumPy/pandas, respectively). Table S2.1 recomputes the mean percentage change for each disparity measure from the five paired runs underlying Table 4 of the main paper. The paired standard error is the sample standard deviation of the five within-pair percentage changes divided by $\sqrt{5}$, and the leave-one-out range drops each pair in turn to assess sensitivity to individual synthetic input replications.

Disparity measure	Mean change (%)	Paired SE	Widening runs	Leave-one-out mean range (%)
MELD at transplant, year 1	+4.8	2.3	5/5	[+2.5, +5.6]
MELD at transplant, year 2	+16.9	5.1	5/5	[+12.5, +19.8]
MELD at transplant, year 3	+20.3	2.6	5/5	[+18.4, +22.0]
Mortality rate, year 1	−7.5	2.3	0/5	[−9.0, −5.6]
Mortality rate, year 2	+7.9	9.5	3/5	[+0.9, +13.8]
Mortality rate, year 3	+1.1	7.8	3/5	[−5.5, +5.5]
New listings: transplant chance	+10.4	1.6	5/5	[+9.1, +11.5]
New listings: death risk	+4.1	3.3	3/5	[+2.4, +6.4]
Already waiting: transplant chance	−0.8	2.6	2/5	[−2.7, +1.2]
Already waiting: death risk	−6.6	3.7	1/5	[−8.6, −3.1]

Table S2.1. Dispersion and leave-one-out sensitivity of the LivSim comparisons in Table 4 of the main paper. Positive changes widen the cross-DSA standard deviation; negative changes narrow it. Statistics describe the five shipped synthetic input replications and do not quantify uncertainty about actual patients or clinical policy effects.

The mean change in transplant-chance disparity for patients already waiting crosses zero under leave-one-out omission ([−2.7%, +1.2%]), whereas widening among new listings retains its sign in every individual replication and every leave-one-out mean ([+9.1%, +11.5%]). Third-year mortality disparity likewise crosses zero under omission ([−5.5%, +5.5%]). For the fixed-denominator accounting recount reported in Table 5 of the main paper, empty DSAs are omitted under LivSim's convention and the nonempty area set stays identical across policies in each pair-year. `livsim_fixed_denominators.py` saves all 30 reconstructed published values, common-denominator accounting comparisons, and input hashes in `evidence/LIVSIM_FIXED_DENOMINATORS.json`; `review_results/LIVSIM_DENOMINATOR_SHIFTS.json` retains the exact unrounded paired shifts.

Supplement S3. Formal Proofs, Triage Sensitivity Grid, Reproducibility, ODD Specification, and Executable Record Export

S3.1 Formal proofs of Theorem 1 and Proposition 1

Proof of Theorem 1 (Necessary and sufficient conditions for sign reversal and pairwise ranking reversal). Let all compared policies be evaluated on a shared exogenous problem instance with common policy-invariant reference $F > 0$, positive realised values $V_p > 0$, and positive moving references $B_p > 0$. Recall $\Phi_p = V_p/F$ and $E_p = V_p/B_p$.

Part (i) — Sign reversal against baseline. Because $F > 0$ is invariant across arms, $\Phi_1 - \Phi_0 = (V_1 - V_0)/F$ shares the strict sign of $V_1 - V_0$. By strict monotonicity of the natural logarithm on $(0, \infty)$:

$$V_1 < V_0 \Leftrightarrow \frac{V_0}{V_1} > 1 \Leftrightarrow \log \frac{V_0}{V_1} > 0,$$

and using equation (1) of the main text,

$$E_1 > E_0 \Leftrightarrow \frac{E_1}{E_0} > 1 \Leftrightarrow \log \frac{E_1}{E_0} = \log \frac{V_1}{V_0} - \log \frac{B_1}{B_0} > 0 \Leftrightarrow \log \frac{B_0}{B_1} > \log \frac{V_0}{V_1}.$$

Combining both equivalences gives $V_1 < V_0$ and $E_1 > E_0$ if and only if $\log(B_0/B_1) > \log(V_0/V_1) > 0$ (equivalently $B_1/B_0 < V_1/V_0 < 1$). Reversing the inequalities yields $V_1 > V_0$ and $E_1 < E_0$ if and only if $\log(B_1/B_0) > \log(V_1/V_0) > 0$ (equivalently $B_1/B_0 > V_1/V_0 > 1$).

Part (ii) — Pairwise policy ranking reversal. Let policies A and B be evaluated against a shared baseline 0 on the same exogenous instance with fixed reference $F > 0$. Because E_0 and Φ_0 are common constants across the two arms, $\Delta E_A > \Delta E_B \Leftrightarrow E_A > E_B$ and $\Delta\Phi_A < \Delta\Phi_B \Leftrightarrow \Phi_A < \Phi_B \Leftrightarrow V_A < V_B$. Applying the same logarithmic equivalence directly to the pair (A, B) gives

$$V_A < V_B \Leftrightarrow \log\frac{V_B}{V_A} > 0, \qquad E_A > E_B \Leftrightarrow \frac{V_A/B_A}{V_B/B_B} > 1 \Leftrightarrow \log\frac{B_B}{B_A} > \log\frac{V_B}{V_A}.$$

Hence $E_A > E_B$ while $V_A < V_B$ if and only if $\log(B_B/B_A) > \log(V_B/V_A) > 0$. Because neither inequality restricts V_0 or B_0, the condition can hold simultaneously with $V_B > V_A > V_0$ (both policies strictly beat baseline on realised value) and $E_A > E_B > E_0$ (both improve on the moving benchmark). Across R matched replications $r = 1, ..., R$, checking both the per-replication reversal frequency $\sum_{r=1}^{R} \mathbf{1}\{\log(B_{B,r}/B_{A,r}) > \log(V_{B,r}/V_{A,r}) > 0\}$ and the pooled log difference $\log(\sum_r B_{B,r}/\sum_r B_{A,r}) > \log(\sum_r V_{B,r}/\sum_r V_{A,r}) > 0$ guards against sign flips induced purely by cross-replication weighting. ▫

Proof of Proposition 1 (Finite-horizon open-cohort censoring bias identity). Under matched exogenous arrivals on $[0, T]$, the total arrival count N, the completed-cohort count $N^{\text{comp}} = |\mathcal{C}|$ ($\mathcal{C} = \{i : t_i \le T - H\}$), the arrival-time counts N_t for $t > T - H$, the weights $\omega_t = N_t/N$, and the late-arrival share $\Omega = \sum_{t>T-H} \omega_t = (N - N^{\text{comp}})/N = 1 - N^{\text{comp}}/N$ are identical across all policies. Let $I_{p,i} = \mathbf{1}\{\text{entrant } i \text{ exits unserved by } T \text{ under policy } p\}$. Decomposing the full-run numerator over $\mathcal{C}$ and $\{t > T - H\}$ yields

$$U_p^{\text{open}} = \frac{1}{N}\left(\sum_{i\in\mathcal{C}} I_{p,i} + \sum_{t>T-H}\sum_{i:t_i=t} I_{p,i}\right) = \frac{N^{\text{comp}}}{N} U_p^{\text{comp}} + \sum_{t>T-H} \frac{N_t}{N} U_{p,t}^{\text{cens}}.$$

Substituting $N^{\text{comp}}/N = 1 - \sum_{t>T-H} \omega_t = 1 - \Omega$ immediately gives equation (2):

$$U_p^{\text{open}} = U_p^{\text{comp}} + \sum_{t>T-H} \omega_t\big(U_{p,t}^{\text{cens}} - U_p^{\text{comp}}\big) = U_p^{\text{comp}} + \Omega\big(\bar{U}_p^{\text{cens}} - U_p^{\text{comp}}\big),$$

where $\bar{U}_p^{\text{cens}} = \Omega^{-1} \sum_{t>T-H} \omega_t U_{p,t}^{\text{cens}}$ is the late-window unserved rate observed by T.

Subtracting equation (2) for Policy B from Policy A (using policy-invariance of ω_t and Ω) yields

$$U_A^{\text{open}} - U_B^{\text{open}} = (1 - \Omega)(U_A^{\text{comp}} - U_B^{\text{comp}}) + \sum_{t>T-H} \omega_t\big(U_{A,t}^{\text{cens}} - U_{B,t}^{\text{cens}}\big), \qquad \text{(S3.1)}$$

whose final sum equals $\Omega(\bar{U}_A^{\text{cens}} - \bar{U}_B^{\text{cens}})$. If $U_A^{\text{comp}} = U_B^{\text{comp}}$, the first term vanishes and $U_A^{\text{open}} - U_B^{\text{open}} = \sum_{t>T-H} \omega_t (U_{A,t}^{\text{cens}} - U_{B,t}^{\text{cens}})$, proving that hazard-timing differences in $(T - H, T]$ shift U^{open} even when completed-cohort access is identical. When $U_A^{\text{comp}} < U_B^{\text{comp}}$ (so $U_B^{\text{comp}} - U_A^{\text{comp}} > 0$), equation (S3.1) is strictly positive ($U_A^{\text{open}} > U_B^{\text{open}}$) if and only if

$$\sum_{t>T-H} \omega_t\big(U_{A,t}^{\text{cens}} - U_{B,t}^{\text{cens}}\big) > (1 - \Omega)(U_B^{\text{comp}} - U_A^{\text{comp}}) > 0,$$

or equivalently $\frac{\Omega}{1-\Omega}(\bar{U}_A^{\text{cens}} - \bar{U}_B^{\text{cens}}) > U_B^{\text{comp}} - U_A^{\text{comp}} > 0$, establishing equation (3). Finally, writing S_p for total requests served in $[0, T]$ and $W_p(T)$ for unresolved entrants still waiting in the queue at T, conservation of entrants gives $NU_p^{\text{open}} = N - S_p - W_p(T)$, so $N(U_A^{\text{open}} - U_B^{\text{open}}) = (S_B - S_A) + (W_B(T) - W_A(T))$. Whenever two work-conserving policies fill nearly identical total capacity ($S_A \approx S_B$), open-run unserved differences are dominated by the end-of-horizon backlog difference $W_B(T) - W_A(T)$: a policy that holds a larger unresolved queue at T appears artificially superior under open-run accounting. ▫

S3.2 Canonical triage DES specification and offered-load sensitivity grid

The canonical multi-class priority-triage discrete-event simulation (`triage_benchmark.py`, pure Python standard library) runs $T = 120$ discrete periods with $c = 10$ service slots per period, Poisson total arrivals ($\lambda = \rho c$, generated via exact Knuth sampling from `random.Random(seed)`), Urgent class share $p_U = 0.60$ ($v_U \sim U[60, 140]$, deadline length $d_U \sim U\{1, 2, 3, 4\}$), and Routine class share $1 - p_U = 0.40$ ($v_R \sim U[20, 60]$, deadline length $d_R = 30$). All policies share identical CRN arrival streams across 50 matched replications (`seeds 20261118–20261167`), identical fixed upper bound F (sum of top $T \cdot c = 1,200$ arrival valuations), identical moving reference definition B_p (sum over allocated slots of $\max_{i\in \text{ pool}} v_i$), and exact completed entry cohort $\mathcal{C} = \{t \le 90\}$ ($H = 30$, zero unresolved entrants at $T = 120$). Policy A caps Routine queue length at $K = 60$ (rejecting excess Routine arrivals immediately

at arrival) and serves Urgent by HV then admitted Routine FIFO; Policy B never blocks at the gate and serves Urgent by EDF (value tie-break) then Routine by HV.

Table S3.1 reports the 6-point offered-load sensitivity grid ($\rho \in \{1.05, 1.15, 1.25, 1.35, 1.45, 1.55\}$, 50 CRN replications per row; saved in `review_results/TRIAGE_BENCHMARK_SUMMARY.json`). Both Policy A and Policy B outperform Baseline FCFS on fixed value in **50/50** replications at every tested load. the pairwise ranking reversal of Theorem 1(ii) holds in **50/50** replications at every single load, while Proposition 1′s cohort-vs-open access reversal switches on monotonically as congestion causes Policy A's routine gate to bind ($\rho \geq 1.25$).

Offered load ρ	Fixed $\Phi_B - \Phi_A$ (pp; SE)	Moving $E_A - E_B$ (pp; SE)	Completed $U_B^{\text{comp}} - U_A^{\text{comp}}$ (pp; SE)	Open $U_A^{\text{open}} - U_B^{\text{open}}$ (pp; SE)	Thm. 1(ii) rev.	Prop. 1 rev.
1.05	+1.19 (0.07)	+4.06 (0.15)	+3.27 (0.20)	−1.49 (0.11)	50/50	0/50
1.15	+2.51 (0.04)	+5.88 (0.17)	+4.15 (0.09)	−0.21 (0.13)	50/50	17/50
1.25	+3.01 (0.04)	+8.40 (0.19)	+3.97 (0.09)	+1.60 (0.16)	50/50	46/50
1.35	**+3.11 (0.03)**	**+11.29 (0.20)**	**+3.17 (0.09)**	**+3.31 (0.13)**	**50/50**	**50/50**
1.45	+2.86 (0.05)	+14.06 (0.26)	+2.42 (0.09)	+4.59 (0.13)	50/50	50/50
1.55	+2.43 (0.05)	+17.25 (0.22)	+1.45 (0.08)	+5.90 (0.11)	50/50	49/50

Table S3.1. Offered-load sensitivity grid for the canonical two-class priority-triage DES benchmark (50 matched CRN replications per row; seeds `20261118`–`20261167`). Bold row marks the canonical $\rho = 1.35$ specification reported in Table 6 and Figure 1 of the main Technical Note. Positive columns 2 and 3 indicate B leads on fixed value while A leads on the moving benchmark (Theorem 1(ii)); positive columns 4 and 5 indicate A leads on completed-cohort access while B falsely leads on open full-run access (Proposition 1).

S3.3 Housing simulation reproducibility

All housing scripts are relative to `review_sources/housing/`; the root-level fast verification command is `python reproduce_review.py`. The package retains the simulation code of the extended working paper (Loschi 2026b) and adds the main-market and proxy scripts: `realistic_market.py` (main-market benchmark sweep, common full-run and cohort targets, and carry-forward quota), `realistic_tables.py`, `build_figures_v64.py`, `clean_reproduce.py`, `replication/sim_v6.py`, `replication/sim_v6_supply.py`, `replication/sim_v6_search_fixed.py`, `replication/displacement.py`, `replication/welfarist.py`, `replication/breakeven_schedule.py`, `equal_access.py`, `constant_charge.py`, `check_completed_cohorts.py`, `cohort_targets.py`, `carry_share.py`, and `threshold_intervals.py`.

The baseline uses seed 20260824 and twelve replications, advancing the seed by 100 between runs. Common-target exercises tune on seeds $22260824 + 100i$ $(i = 0, ..., 3)$ and evaluate on $23260824 + 100i$ $(i = 0, ..., 15)$; cohort retuning evaluates on sixteen further disjoint seeds. Within each seed, every policy faces identical arrivals, valuations, and listings so oracle denominators align, while reduced-search runs use a separate generator for search draws. In a clean extraction on 21 September 2026 with original-workspace reads blocked, all 1,020 runs behind the constant-charge, cohort, and carry-forward experiments reproduced to floating-point tolerance. The benchmark sweep (144 runs) and cohort audit (80 runs) reproduced all selected fields exactly in Python 3.14.7 with NumPy 2.5.3.

S3.4 ODD model description

Following the ODD protocol (Grimm et al. 2020), the housing model is designed strictly to expose how repeated allocation reshapes the reference queue and does not approximate household search histories in an observed city. Entities are civilian buyer-instances (storing class, valuation, and entry time), institutional buyer-instances (also storing entity ID), and district-period listings. District state tracks waiting buyers, cumulative entity acquisitions, and a trailing six-period resolved exit history. Evaluation routines log realised value V, the contemporaneous pool reference B, and all generated valuations (yielding F). There are no spatial links across districts in the core specification, and all waiting civilians inspect all local listings under full search.

Scheduling and stochastic inputs. Each period executes in fixed order: (1) civilian arrivals, (2) civilian expiry (removing entrants whose elapsed periods reach patience), (3) fresh institutional instances, (4) listings, (5) lagged contention signal calculation, (6) sequential listing allocations, and (7) exit-history update. Each buyer-instance can purchase at most once, and institutional instances expire at the period boundary. Main Poisson arrival/listing and lognormal valuation draws share the primary RNG stream so full-search draw counts stay invariant across policies; transaction-failure and reduced-search diagnostics use separate RNG streams. Under period-reset fair share, institutional purchases are admitted while prior institutional wins in the period are strictly below $s \cdot L$, allowing up to $\lceil s \cdot L \rceil$ wins when $s > 0$.

S3.5 Benchmark, deadline-queue, triage, and executable record verification

- **Two-period counterexample:** `python replication/benchmark_toy.py` enumerates all feasible schedules for Table 1 of the main paper.
- **Deadline queue:** `python replication/deadline_queue.py` regenerates Table 2, all 200 six-slot cases (seeds `20261006`–`20261205`), and the exact DP verification against full enumeration across all 5,832 three-job, three-slot configurations in $\{1, 2, 3\}$. The offline DP maximises value over $(\text{slot}, \text{served-job set})$ states; the independent verifier enumerates complete slot assignments (including idle slots) and rejects repeated jobs or interval violations. Separately, `python deadline_matching_check.py` independently verifies all 200 saved optima via descending-weight greedy selection with augmenting-path bipartite matching on the transversal matroid of schedulable job subsets (Edmonds 1971), storing schedules and comparisons in `review_results/DEADLINE_MATCHING_CHECK.json`.
- **Canonical triage DES benchmark:** `python triage_benchmark.py` runs all 50 matched CRN replications at $\rho = 1.35$ plus the 6-point sensitivity grid ($\rho \in [1.05, 1.55]$), verifies exact Theorem 1(ii) and Proposition 1 residuals ($< 10^{-14}$), and exports all 50 Policy A vs. Policy B contract records (`contract_records/triage_000.json`–`triage_049.json`) including completed-cohort membership hashes (`unresolved = 0`); `python build_triage_figure.py` generates the vector PDF of Figure 1.
- **Portable audit and exported contract records:** `python replication/portable_audit.py` applies `compare_benchmarks.py` across all 272 saved pairs (72 housing stress-test pairs and 200 service pairs). Together, `export_records.py` and `triage_benchmark.py` construct canonical UTF-8 sorted-key JSON payloads (compact separators, `ensure_ascii=False`, `allow_nan=False`, no trailing newline; a local convention rather than RFC 8785) for **256** records under `contract_records/`: 1 exact two-period pair, 200 deadline pairs, 5 fixed-horizon LivSim incident-cohort transplant-count pairs (hashing all 16 shipped synthetic input files against `CONFIG.json` and separating exogenous feasibility from regional sharing), and 50 canonical triage DES pairs (exercising `audit_contract.py`'s completed-cohort verification with `unresolved = 0`). The LivSim count reference is the entry-cohort size (an upper bound ignoring organ capacity, not a feasible clinical optimum), retaining still-waiting patients without making a completed-exit claim. The unchanged `audit_contract.py` CLI passes all 256 records with zero failures.

Supplement S4. Extended Housing Analysis, Stress Tests, and Cohort Retuning

S4.1 The seller's problem and fixed-offer displacement

Suppose a seller holds a unit worth P, while a cash offer C closes at horizon τ_C and a financed offer F closes with probability $1 - q$ ($0 \le q < 1$) at $\tau_F > \tau_C$. If the financed sale fails, the seller incurs relisting loss δ and eventually sells at P. A risk-neutral seller with holding cost k per period is indifferent when

$$(1 - q)(F - k\tau_F) + q(P - \delta - k\tau_F) = C - k\tau_C. \qquad \text{(S4.1)}$$

Setting $P = C$, $d = \delta/C$, and $\kappa = k(\tau_F - \tau_C)/C$, the risk-justified discount on the financed price, $\pi^* = (F - C)/F$, is

$$\pi^* = \frac{qd + \kappa}{1 - q + qd + \kappa}. \qquad \text{(S4.2)}$$

Only a discount exceeding π^* represents a candidate distortion beyond completion risk and carry cost; neither a transaction price gap nor a mortgage denial rate identifies π^* without local data on failed accepted offers. Both cost channels can exceed mechanical relisting costs due to seller loss aversion (Genesove and Mayer 2001) and cyclical time-to-sell (Head, Lloyd-Ellis and Sun 2014).

Consider one listing with fixed offers and eligibility where targeted buyer j has valuation v_j and certainty factor c_j, and untargeted buyer i has v_i and c_i. Moving the unit from j to i changes realised valuation by $-G$ where $G = v_j - v_i$.

Proposition S4.1 (fixed-offer selection). *Let $a = v_j c_j$ be the highest effective offer among targeted buyers and $b = v_i c_i$ the highest among untargeted buyers. Suppose $a \ge b > 0$, ties go to the targeted group, and offers and eligibility are fixed. Multiplying every targeted offer by $(1 - \theta)$, with $0 < \theta < 1$, changes the winning group if and only if*

$$r = 1 - \frac{b}{a} < \theta. \qquad \text{(S4.3)}$$

If $c_j > c_i$ and $\theta \le 1 - c_i/c_j$, every such displacement has $v_i > v_j$.

Proof. Scaling all targeted offers by $(1-\theta)$ preserves their internal ranking, so the best targeted effective offer after the charge is $(1-\theta)a$. The untargeted group wins if and only if $b > (1-\theta)a$, which rearranges to (S4.3). Any displacement therefore satisfies $v_i/v_j > (1-\theta)c_j/c_i \geq 1$ under $\theta \leq 1 - c_i/c_j$. ▫

With private valuations and asymmetric bidders, first-price shading could alter equilibrium winners (Maskin and Riley 2000; Hafalir and Krishna 2008), so (S4.3) characterises displacement strictly for fixed offers. Moreover, selecting listings by normalised margin r does not determine average valuation cost: on two equally likely listings with $(v_j, v_i) \in \{(10,000, 9,700), (100, 80)\}$ and $(c_j, c_i) = (1, 0.88)$, a 20% charge flips only the first listing (valuation loss 300), whereas excluding j with probability $1/2$ yields the same expected displacements at conditional mean loss 160. In the stress-test simulation, removing the active instrument at each listing gives mean G of $+143,789$ for the cap, $+22,188$ for the fair share, and $-12,905$ for the duty, with 9.0%, 32.8%, and 80.4% of displacements shifting units to higher-valuation buyers; these local differences cannot be summed into aggregate effects because displaced buyers may transact later.

S4.2 Benchmark stress tests and break-even schedule

In the high-participation stress test (42 institutional bids per district-period, 58.1% unconstrained share, baseline $w = 12\%$), the contemporaneous reference B averages roughly 521,000 per unit without intervention, 562,000 under the cap, and 506,000 under the duty (an 11% spread). On the fixed oracle, the duty begins adding value at $w \approx 11.7\%$ (95% CI [11.5, 11.8]), versus 5.4% ([5.4, 5.5]) on the moving benchmark (Figure S4.1).

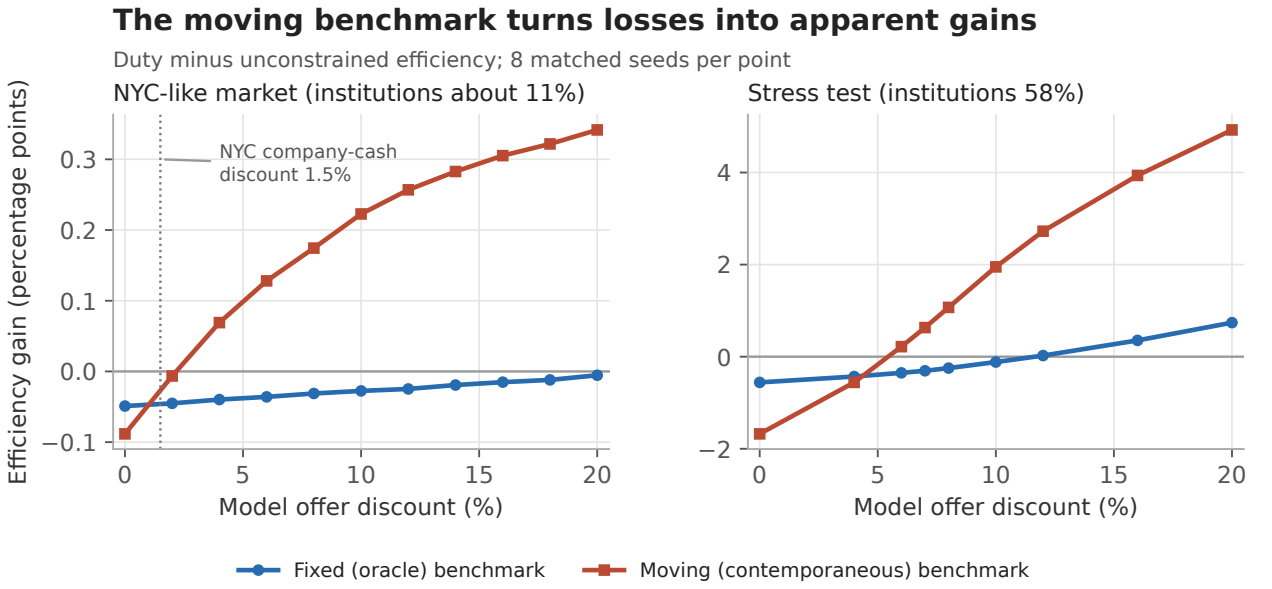

Figure S4.1. Duty efficiency gain on the fixed (oracle) and moving (contemporaneous) benchmarks across offer discounts in the main market (left, Table 7) and the stress test (right). Lines join simulation means over 8 matched seeds.

Institutional share at $w = 12\%$	Interpolated break-even w	95% interval
12.6%	none up to 20%	–
22.4%	15.9%	[15.6, 16.1]
29.9%	14.2%	[13.8, 14.7]
40.2%	12.5%	[12.2, 12.7]
46.8%	12.1%	[11.8, 12.3]
52.8%	11.9%	[11.8, 12.1]
58.1%	11.7%	[11.5, 11.8]
63.5%	11.9%	[11.7, 12.0]
68.3%	11.8%	[11.7, 11.9]
73.7%	12.0%	[11.8, 12.1]
79.9%	12.3%	[12.2, 12.4]

Table S4.1. Offer discount at which the duty stops losing value on the fixed benchmark across eleven institutional-capacity settings ($\tau_{\max} = 0.22$). Row 1 is the main market; rows 2–11 are the stress-test capacity sweep. Crossings are linearly interpolated between 2-pp grid nodes (8 replications per node); 95% percentile intervals use 4,000 bootstrap resamples of matched seeds (`threshold_intervals.py`).

Table S4.2 decomposes the stress-test sweep into pooled value change, pooled moving-reference change, and both ratio gains (underlying Figure 2 and Section 1.1 of the main paper). Across the nine nodes, 33 of 72 matched pairs exhibit opposite signs (all 8 of 8 seeds at 6%, 7%, 8%, and 10%, plus 1 of 8 at 12%). Largest log-identity residual is approximately 2.31×10^{-16}, and pooled versus mean-per-seed moving-ratio changes differ by less than 0.0003 pp across all nine nodes.

Offer discount	Value change (%)	B change (%)	Gain vs F (pp)	Gain vs B (pp)	Opposite signs
0%	−0.561	+1.134	−0.559	−1.676	0/8
4%	−0.433	+0.131	−0.431	−0.558	0/8
6%	−0.352	−0.573	−0.351	+0.218	8/8
7%	−0.307	−0.946	−0.306	+0.632	8/8
8%	−0.249	−1.332	−0.248	+1.071	8/8
10%	−0.117	−2.096	−0.117	+1.950	8/8
12%	+0.024	−2.754	+0.024	+2.725	1/8
16%	+0.356	−3.726	+0.353	+3.936	0/8
20%	+0.749	−4.483	+0.740	+4.925	0/8

Table S4.2. Stress-test matched-seed sweep (8 seeds per offer discount). Value and B changes are percentages from pooled totals; ratio gains are percentage-point differences of mean per-seed ratios. The final column counts opposite signs for realised value and moving-ratio change within a seed.

S4.3 Main-market cohort retuning

Figure 3 of the main Technical Note visualises the main-market common full-run access targets (Table 8), and Table S4.3 reports the corresponding results when every instrument is retuned directly to completed-cohort targets of 0.5 and 1 percentage point and evaluated on sixteen further disjoint replications.

Target (pp)	Instrument	Setting	Cohort gain (pp; SE)	Full-run gain (pp; SE)	Value index (SE)
0.5	Static cap	75	0.44 (0.055)	0.18 (0.022)	99.60 (0.019)
0.5	Fair share, period reset	0.1016	0.56 (0.035)	0.40 (0.018)	99.89 (0.005)
0.5	Contention duty	0.2542	0.50 (0.020)	0.23 (0.006)	99.94 (0.002)
0.5	Constant charge	0.0297	0.52 (0.029)	0.36 (0.012)	99.92 (0.002)
0.5	Fair share, carry-forward	0.1040	0.51 (0.053)	0.44 (0.034)	99.87 (0.010)
1	Static cap	70	0.88 (0.077)	0.35 (0.032)	99.42 (0.020)
1	Fair share, period reset	0.0938	0.95 (0.049)	0.67 (0.025)	99.81 (0.006)
1	Contention duty	0.5307	1.07 (0.040)	0.50 (0.017)	99.86 (0.004)
1	Constant charge	0.0561	1.07 (0.050)	0.72 (0.017)	99.84 (0.004)
1	Fair share, carry-forward	0.0974	1.05 (0.060)	0.80 (0.038)	99.76 (0.011)

Table S4.3. Main-market instruments tuned to common completed-cohort access targets (4 tuning seeds, 16 disjoint evaluation seeds). Each price rule retains slightly more value on the population for which it was tuned.

S4.4 Access-population stress tests

In the stress test, tuning instruments to 5- and 10-point full-run access targets yields Tables S4.4–S4.5 and Figure S4.2. On the completed cohort (averaging 4,061 entrants per replication, 60.1% unserved unconstrained vs. 41.9% full-run), full-run accounting understates gains by 1.4× (fair share) to 2.5× (cap) and shifts rankings. Retuning to 5- and 10-point cohort targets requires a cap of 385 (vs. 336) at 5 pp and flips the charge comparison at 10 pp: the duty retains +0.032 value-index points over the constant charge (paired SE 0.0033) at virtually identical cohort gains (within 0.03 pp, SE 0.044) while the constant charge delivers +1.6 pp more full-run access (SE 0.044); near 5 pp the gap (+0.0012, SE 0.0017) is within Monte Carlo noise.

Target (pp)	Instrument	Setting	Access gain (pp; SE)	Value index (SE)	Constant − duty, value (SE)
5	Static cap	336	5.42 (0.121)	97.46 (0.048)	–
5	Fair share	0.5000	5.78 (0.086)	99.99 (0.007)	–
5	Signal duty	0.2115	5.03 (0.031)	100.03 (0.005)	–
5	Constant charge	0.0779	4.97 (0.041)	100.08 (0.005)	+0.051 (0.003)
10	Static cap	273	10.64 (0.150)	95.73 (0.061)	–
10	Fair share	0.4121	10.14 (0.100)	99.63 (0.014)	–
10	Signal duty	0.4657	10.05 (0.070)	99.32 (0.009)	–
10	Constant charge	0.1466	9.79 (0.034)	99.68 (0.008)	+0.361 (0.008)

Table S4.4. Stress-test instruments tuned to common full-run access targets (4 tuning seeds, 16 held-out seeds). Final column gives the paired delta-method value difference between constant charge and duty.

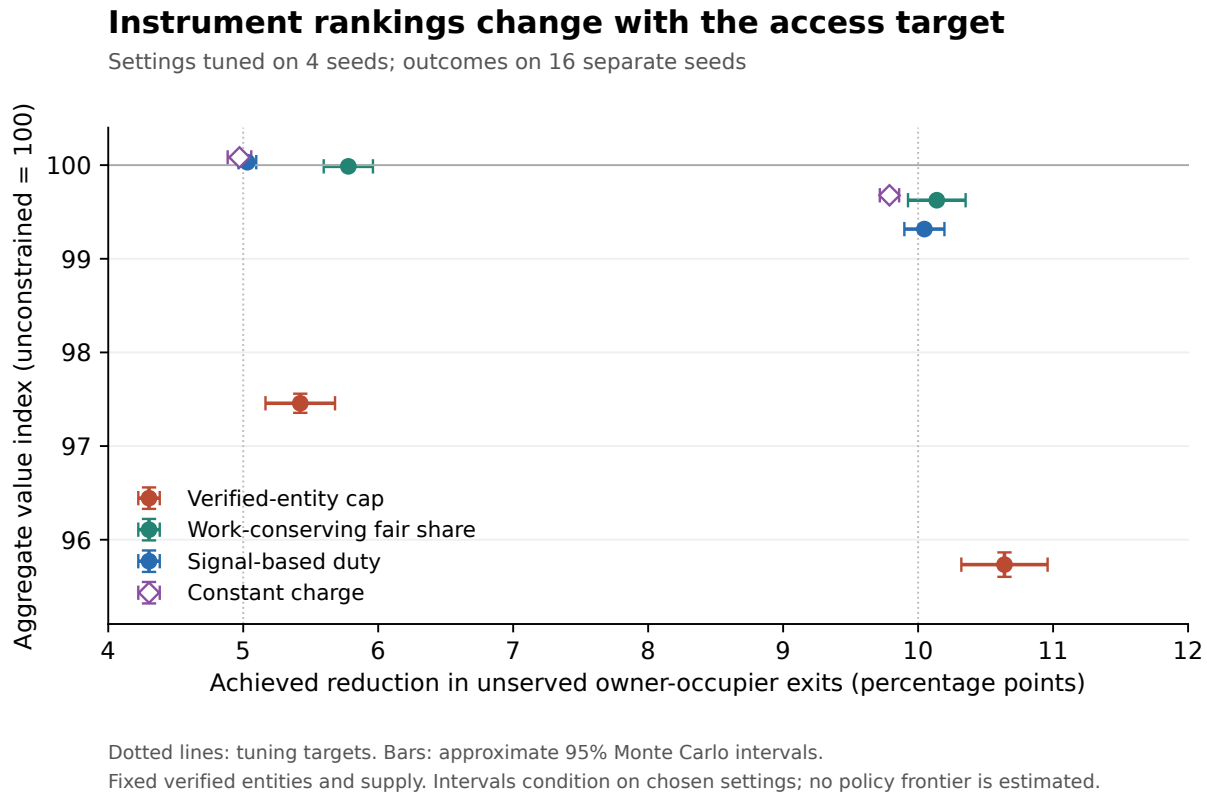

Figure S4.2. Stress-test results tuned to common full-run access targets and evaluated on 16 held-out seeds. Bars show approximate 95% Monte Carlo intervals conditional on settings.

Regime	Setting	Full-run gain (pp)	Cohort gain (pp; SE)	Cohort unserved (%)
Unconstrained	–	–	–	60.09
Static cap	336	5.42	13.53 (0.279)	46.56
Fair share	0.5000	5.78	7.91 (0.134)	52.18
Signal duty	0.2115	5.03	9.34 (0.072)	50.75
Constant charge	0.0779	4.97	7.15 (0.065)	52.94
Static cap	273	10.64	26.44 (0.318)	33.65
Fair share	0.4121	10.14	13.97 (0.164)	46.12
Signal duty	0.4657	10.05	18.04 (0.140)	42.05
Constant charge	0.1466	9.79	14.03 (0.091)	46.06

Table S4.5. Stress-test completed-cohort access at the settings of Table S4.4 without retuning (cohort averages 4,061 entrants per replication).

S4.5 Stress-test baseline regimes and numerical sensitivities

Regime	OO share	Inst. share	Starvation	Value index	Oracle eff.	Contemp. eff.
Unconstrained	34.0%	58.1%	41.9%	100.00	99.5%	95.4%
Static per-entity cap	75.8%	6.3%	8.7%	88.72	88.3%	78.5%
Work-conserving fair share	54.5%	32.2%	24.7%	98.22	97.7%	95.2%
Contention-triggered duty	41.1%	49.4%	36.6%	100.03	99.5%	98.1%
Combined (share + duty)	54.7%	32.1%	24.6%	98.20	97.7%	96.3%

Table S4.6. Allocation regimes in the stress test (means across 12 replications).

In Table S4.6, the duty lowers starvation from 41.9% to 36.6% with a minor value gain (+0.026 index points, paired SE 0.0054), while the fair share and cap trade larger starvation drops for 1.8% and 11.3% value losses. Under distributional weights $W(\omega) = \omega V_{\mathrm{OO}} + \sqrt{\omega} V_{\mathrm{SI}} + V_{\mathrm{I}}$ (`replication/welfarist.py`), the fair share overtakes the duty above $\omega \approx 1.16$, the cap breaks even against no intervention near $\omega \approx 1.35$, and the cap leads above $\omega \approx 1.66$.

Additional sensitivities from the extended working paper (Loschi 2026b): (i) costless legal-entity splitting neutralises the stress-test cap completely at 25 entities, and while starvation rises, 99th-percentile completed waits *fall* due to survivor selection; (ii) unit-elastic listing supply widens the cap's value loss to 17.3% (starvation 16.0%) and gives the duty a 2.0-point value loss; (iii) spatial budget reallocation proportional to $(1 - \tau_{\max} s_d)^{\lambda}$ across $\lambda \in [0, 8]$ shifts the duty's value index from 99.60 to 100.22 (starvation 36.5%–36.7%); (iv) doubling owner-occupier arrivals to 34 per district-period lowers unconstrained institutional share to 38% and allows the fair share (100.20) to overtake the duty (100.09) while the cap remains costliest (97.3); (v) 10% financed-sale failure preserves rankings and fixed break-even (11.8% vs. 11.7%) while raising failed sales by 21.6% under the duty and more than doubling them under the cap; (vi) search inspection probabilities of 0.5 and 0.25 preserve rankings; (vii) sweeping mean institutional valuation from 340,000 to 410,000 shifts the duty's index by ≤ 0.06 points and expands the cap's loss from 8.6% to 16.0%; (viii) lowering institutional share to 22.5% leaves duty and fair-share starvation cuts at 0.7 and 0.1 pp; and (ix) sweeping $\tau_{\max}$ peaks at $\tau_{\max} = 0.11$ (100.12 vs. 100.02 at 0.22), while at $w = 5.6\%$ all tested ceilings (0.11, 0.22, 0.44) lose value.

Supplement S5. New York City Proxies, Parameters, Baseline Defaults, and Implementation Notes

S5.1 New York City proxy evidence and scenarios

The main-market scale proxies come from Loschi (2026a, Version 5.5) and the linked transaction dataset (Loschi 2026d), where company cash buyers account for 19,019 of 166,227 ordinary-deed house sales of at least $100,000 (11.4%), all company buyers account for 22.8%, and the company-cash log-price gap is 1.53 log points (95% interval [−1.79, 5.44]; approximate cash discount 1.5%, upper endpoint 5.3%). Company status and unmatched mortgages do not identify institutional buyers or cash financing cleanly, so Table S5.1 and Table S5.2 serve strictly as scale checks.

Market	Log-price gap, 100β	95% interval	Approx. cash discount
One- to three-family houses, headline	9.25	[7.17, 11.35]	8.8%
Houses, excluding same-surname deeds	4.87	[2.69, 6.76]	4.8%
Houses, cash buyer is a company	1.53	[−1.79, 5.44]	1.5%
Houses, cash buyer is another party	13.77	[10.96, 15.98]	12.9%
Houses, other party, excluding same-surname deeds	7.11	[4.48, 9.12]	6.9%
Condominiums	0.83	[0.06, 1.61]	0.8%
Co-operatives	−1.42	[−3.70, 0.86]	−1.4%

Table S5.1. Financing-related price gaps in New York City, 2016–2025 (log points, source Loschi 2026a, Version 5.5). Cash discounts follow $100[1 - \exp(-\beta)]$. *House intervals use parcel bootstraps; co-operatives cluster by building.*

Scenario	Discount / model share (%)	Unserved: full / cohort (%)	Cap	Fair share	Duty	Constant
Company cash	1.5 / 11.0	9.8 / 14.9	−1.77 / −1.36	0.00 / 0.00	−0.20 / −0.05	−0.24 / −0.05
Upper gap	5.3 / 11.5	10.6 / 16.0	−2.15 / −1.47	0.00 / 0.00	−0.21 / −0.04	−0.26 / −0.04
All companies	1.5 / 25.0	18.8 / 27.2	−11.29 / −5.16	−0.46 / −0.06	−1.09 / −0.18	−1.21 / −0.17
Headline gap	8.8 / 12.1	11.1 / 16.8	−2.53 / −1.56	0.00 / 0.00	−0.20 / −0.03	−0.25 / −0.03
Stress test	12.0 / 58.1	41.9 / 59.8	−33.16 / −11.28	−17.14 / −1.78	−5.24 / +0.03	−4.85 / +0.10

Table S5.2. NYC proxy scenarios and stress test across 12 matched seeds. Instrument cells report full-run starvation change (pp) / realised-value change (%) relative to unconstrained. Under Upper gap and Headline gap, the moving benchmark yields apparent duty gains of +0.11 and +0.19 pp while the fixed oracle records losses of −0.04 and −0.03 pp. New York mortgage recording taxes (averaging 1.41% for houses and 1.28% for condominiums at headline borrower rates; Loschi 2026c) lie outside the model.

S5.2 Housing parameter details and default instrument comparisons

Civilian arrivals average 17 owner-occupiers and 4 small investors per district-period against 20 listings, with lognormal valuations centred near 400,000 ($\sigma = 0.34$) and 372,000 ($\sigma = 0.30$) and patience 36 and 24 periods. Institutional valuations centre near 366,000 ($\sigma = 0.18$). A financed price premium p corresponds to $w = p/(1 + p)$

and a log-price gap β to $w = 1 - \exp(-\beta)$. The trailing six-period duty signal averages 0.094 unconstrained, 0.092 under the duty, and 0.076 under the cap in the main market (and 0.406, 0.341, and 0.078 in the stress test).

Regime	OO share	Inst. share	Unserved, full run	Unserved, cohort	Value index	Oracle eff.	Contemp. eff.
Unconstrained	72.3%	11.0%	9.83%	14.87%	100.00	99.54%	99.93%
Static per-entity cap	76.1%	6.3%	8.06%	10.47%	98.64	98.18%	94.53%
Work-conserving fair share	72.3%	11.0%	9.83%	14.87%	100.00	99.54%	99.93%
Contention-triggered duty	72.7%	10.5%	9.63%	14.42%	99.95	99.49%	99.91%
Constant charge	72.6%	10.6%	9.60%	14.50%	99.95	99.49%	99.95%

Table S5.3. Main-market allocation regimes at default settings (means of 12 replications; constant charge $h = 0.020$ matched to duty's mean scheduled haircut). The default 30% fair share does not bind because three entities submit at most three bids per district-period.

S5.3 Related allocation literature and administrative requirements

The housing diagnostic connects to cash-premium evidence (Reher and Valkanov 2024; Han and Hong 2024), non-valuation market-design advantages (Che and Gale 1998; Budish, Cramton and Shim 2015), segmented housing search (Piazzesi, Schneider and Stroebel 2020), transfer-tax notches (Kopczuk and Munroe 2015), second-best taxation and price-vs-quantity banking (Lipsey and Lancaster 1956; Weitzman 1974, 2018; Akbarpour, Dworczak and Kominers 2024), welfare weighting (Atkinson 1970; Bertsimas, Farias and Trichakis 2012), dynamic queueing and kidney exchange (Bloch and Cantala 2017; Thakral 2019; Agarwal et al. 2019; Baccara, Lee and Yariv 2020; Sönmez, Ünver and Yenmez 2020; Waldinger 2021; Leshno 2022), and housing ABMs (Tesfatsion 2006; Geanakoplos et al. 2012; Yun and Moon 2020; Gamal et al. 2024).

Administratively, real-world counterparts of the stylized instruments face steep information requirements omitted from the diagnostic: (1) a contention duty requires an independent, manipulation-resistant exit registry (analogous to Björkegren, Blumenstock and Knight 2026) that routine deed records do not provide; (2) per-entity caps require statutory beneficial-ownership aggregation to prevent entity splitting; (3) the work-conserving release rarely binds when queues persist; and (4) legal classification (cash payment vs. institutional status vs. co-operative coverage, as illustrated by the May 2026 New York cash-home tax proposal; Bloomberg News 2026) determines statutory incidence and cross-border substitution.

Supplement References

Agarwal, N., Ashlagi, I., Azevedo, E., Featherstone, C. R., & Karaduman, Ö. (2019). Market failure in kidney exchange. *American Economic Review*, 109(11), 4026–4070.

Akbarpour, M., Dworczak, P., & Kominers, S. D. (2024). Redistributive allocation mechanisms. *Journal of Political Economy*, 132(6), 1831–1875.

Atkinson, A. B. (1970). On the measurement of inequality. *Journal of Economic Theory*, 2(3), 244–263.

Baccara, M., Lee, S., & Yariv, L. (2020). Optimal dynamic matching. *Theoretical Economics*, 15(3), 1221–1278.

Bertsimas, D., Farias, V. F., & Trichakis, N. (2012). On the efficiency–fairness trade-off. *Management Science*, 58(12), 2234–2250.

Björkegren, D., Blumenstock, J. E., & Knight, S. (2026). Manipulation-robust prediction. *American Economic Review*, 116(9), 3263–3293.

Bloch, F., & Cantala, D. (2017). Dynamic assignment of objects to queuing agents. *American Economic Journal: Microeconomics*, 9(1), 88–122.

Bloomberg News (2026). NY plans new tax on NYC homes over $1 million bought with cash (14 May); NYC cash home purchases tax is likely to be dropped from New York state budget (21 May); New York lawmakers approve second-home tax in state budget (27 May). Bloomberg.

Budish, E., Cramton, P., & Shim, J. (2015). The high-frequency trading arms race: Frequent batch auctions as a market design response. *Quarterly Journal of Economics*, 130(4), 1547–1621.

Che, Y.-K., & Gale, I. (1998). Standard auctions with financially constrained bidders. *Review of Economic Studies*, 65(1), 1–21.

Edmonds, J. (1971). Matroids and the greedy algorithm. *Mathematical Programming*, 1, 127–136. https://doi.org/10.1007/BF01584082

Furtado, B. A. (2022). PolicySpace2: Modeling markets and endogenous public policies. *Journal of Artificial Societies and Social Simulation*, 25(1), 8. https://doi.org/10.18564/jasss.4742

Gamal, Y., Elsenbroich, C., Gilbert, N., Heppenstall, A., & Zia, K. (2024). A behavioural agent-based model for housing markets: Impact of financial shocks in the UK. *Journal of Artificial Societies and Social Simulation*, 27(4), 5. https://doi.org/10.18564/jasss.5518

Geanakoplos, J., Axtell, R., Farmer, J. D., Howitt, P., Conlee, B., Goldstein, J., Hendrey, M., Palmer, N. M., & Yang, C.-Y. (2012). Getting at systemic risk via an agent-based model of the housing market. *American Economic Review*, 102(3), 53–58.

Genesove, D., & Mayer, C. (2001). Loss aversion and seller behavior: Evidence from the housing market. *Quarterly Journal of Economics*, 116(4), 1233–1260.

Grimm, V., et al. (2020). The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation*, 23(2), 7. https://doi.org/10.18564/jasss.4259

Hafalir, I., & Krishna, V. (2008). Asymmetric auctions with resale. *American Economic Review*, 98(1), 87–112.

Han, L., & Hong, S.-H. (2024). Cash is king? Understanding financing risk in housing markets. *Review of Finance*, 28(6), 2083–2118. https://doi.org/10.1093/rof/rfae025

Head, A., Lloyd-Ellis, H., & Sun, H. (2014). Search, liquidity, and the dynamics of house prices and construction. *American Economic Review*, 104(4), 1172–1210.

Kilambi, V., Bui, K., & Mehrotra, S. (2018). LivSim: An open-source simulation software platform for community research and development for liver allocation policies. *Transplantation*, 102(2), e47–e48. https://doi.org/10.1097/TP.0000000000002000

Kopczuk, W., & Munroe, D. (2015). Mansion tax: The effect of transfer taxes on the residential real estate market. *American Economic Journal: Economic Policy*, 7(2), 214–257.

Leshno, J. D. (2022). Dynamic matching in overloaded waiting lists. *American Economic Review*, 112(12), 3876–3910.

Lipsey, R. G., & Lancaster, K. (1956). The general theory of second best. *Review of Economic Studies*, 24(1), 11–32.

Loschi, P. (2026a). *Who Gets the House? Related-Party Transfers and the Cash–Mortgage Price Gap in New York City* (Version 5.5) [working paper and replication package]. Zenodo. https://doi.org/10.5281/zenodo.24343848

Loschi, P. (2026b). *The Execution-Certainty Wedge: Allocation, Distribution, and the Design of Housing-Market Instruments* (Version 5.1, extended version) [working paper and replication package]. Zenodo. https://doi.org/10.5281/zenodo.22977667

Loschi, P. (2026c). *The Shape of the Tax: Round Prices and Bunching at New York City's 2019 Transfer-Tax Notches* (Version 3.10) [working paper]. Zenodo. https://doi.org/10.5281/zenodo.22925301

Loschi, P. (2026d). *NYC residential sales linked to ACRIS deeds and mortgages, 2016–2025* (Version 1.2.0) [dataset]. Zenodo. https://doi.org/10.5281/zenodo.22977532

Maskin, E., & Riley, J. (2000). Asymmetric auctions. *Review of Economic Studies*, 67(3), 413–438.

Piazzesi, M., Schneider, M., & Stroebel, J. (2020). Segmented housing search. *American Economic Review*, 110(3), 720–759.

Reher, M., & Valkanov, R. (2024). The mortgage-cash premium puzzle. *The Journal of Finance*, 79(5), 3149–3201.

Sönmez, T., Ünver, M. U., & Yenmez, M. B. (2020). Incentivized kidney exchange. *American Economic Review*, 110(7), 2198–2224.

Tesfatsion, L. (2006). Agent-based computational economics: A constructive approach to economic theory. In L. Tesfatsion & K. L. Judd (Eds.), *Handbook of Computational Economics*, Vol. 2, 831–880. Elsevier.

Thakral, N. (2019). Matching with stochastic arrival. *AEA Papers and Proceedings*, 109, 209–212.

Waldinger, D. (2021). Targeting in-kind transfers through market design: A revealed preference analysis of public housing allocation. *American Economic Review*, 111(8), 2660–2696.

Weitzman, M. L. (1974). Prices vs. quantities. *Review of Economic Studies*, 41(4), 477–491.

Weitzman, M. L. (2018). *Prices or Quantities Dominate Banking and Borrowing*. NBER Working Paper 24218.

Yun, T.-S., & Moon, I.-C. (2020). Housing market agent-based simulation with loan-to-value and debt-to-income. *Journal of Artificial Societies and Social Simulation*, 23(4), 5. https://doi.org/10.18564/jasss.4410