

SPARSE TOKEN INTERVENTIONS FOR ROUTE CONTROL IN AUTOREGRESSIVE TRAFFIC SIMULATORS

Anonymous authors

Paper under double-blind review

ABSTRACT

Autoregressive traffic simulators generate diverse futures, but a requested route can remain unlikely under ordinary sampling. We ask whether a few early token interventions can steer a frozen simulator while leaving subsequent motion to the model. Sparse Goal Steering (SGS) uses route geometry to restrict a target agent’s motion-token support, searches short intervention schedules, and then releases control. It counts a replacement only when the chosen token differs from the reference decision at the same history, separating direct intervention from downstream effects. In a prospectively reserved evaluation of 512 Waymo source groups, calibration-selected SGS–CEM reaches 34.71% guarded route success under raw sampling, versus 23.92% for native sampling and 28.84% for best-of-16 native selection. Paired source-group gains are 10.79 percentage points (95% CI [9.26,12.36]) and 5.87 points ([4.51,7.27]). In a separate matched 20-group development study, a search-free width-32 mask reaches 26.32% raw success versus 19.87% for its same-job native arm, with 1.80 actual replacements on average. Separate development analyses show that gains persist when arrival must follow release, while continuing the same guidance rule can reduce success. Arbitrary fixed-time endpoint goals remain difficult under the tested interface.

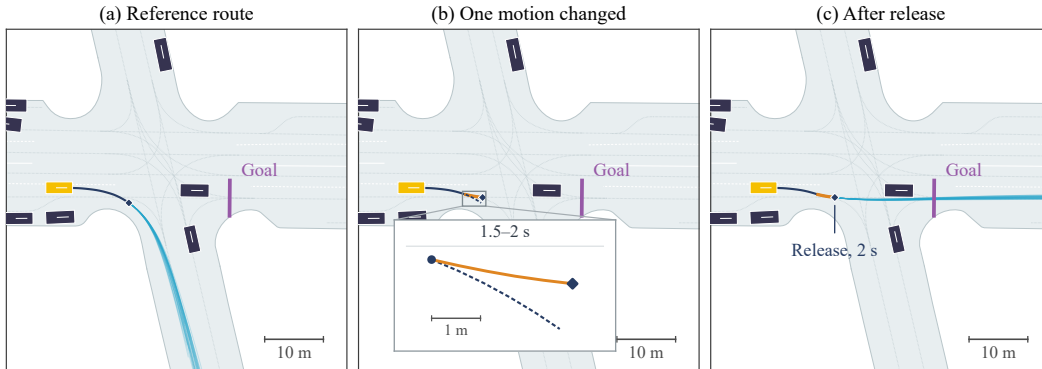


Figure 1: One token replacement redirects subsequent motion. In this selected example, (a) shows the reference route, (b) highlights the replacement at 1.5–2 s, and (c) shows 32 continuations after release at 2 s, of which 31 reach the goal by 5 s. The inset contrasts the reference motion (dashed) with the replacement (orange). Vehicle footprints mark observed initial states.

1 INTRODUCTION

Testing a traffic interaction may require a vehicle to take a different exit while surrounding agents respond. Autoregressive simulators generate interactive futures from driving logs (Phillion et al., 2024; Wu et al., 2024), yet a requested route may remain rare under ordinary sampling. Existing approaches address control by learning goal-conditioned behavior or optimizing trajectories at test time (Tan et al., 2025; Chen et al., 2026; Yang et al., 2024; Xu et al., 2026).

We investigate a more local control interface. An autoregressive simulator may assign low probability to a requested route even when its motion vocabulary contains decisions that lead toward it. Changing an early decision alters the history from which later motion is generated, so a controller may be able to redirect the rollout without prescribing the rest of the trajectory. We therefore ask whether a few early token replacements can increase route success after guidance ends. Figure 1 illustrates the proposed effect.

We introduce *Sparse Goal Steering* (SGS), a training-free controller that restricts the target agent’s available motion tokens using route geometry while sampling from the frozen simulator’s logits within that support. Search chooses when to apply these restrictions and how many tokens to retain, scoring candidate prefixes through model-generated continuations. After the final restriction, the target returns to reference sampling, while non-target agents use the reference sampler throughout. We assess selected prefixes on fresh continuations excluded from search.

The intervention budget distinguishes direct control from its downstream effects. SGS counts a replacement only when the chosen token differs from the reference decision at the same history under the same randomness. A restriction that leaves the decision unchanged consumes no edit, while an early replacement can alter many later decisions. This accounting lets us ask how much route control is possible with only a few direct interventions.

On Waymo scenes, at most two early replacements improve guarded route success over native sampling and native sample selection under both raw and filtered references. Similar gains across several search rules and a post-hoc arrival analysis show that the benefit extends beyond a particular optimizer and persists after release. A matched cohort comparison further finds that releasing a selected prefix outperforms continued application of the same guidance rule. Together with the much lower success on arbitrary fixed-time endpoints, these results characterize the reach and limits of sparse route control in the tested SMART-family simulator.

2 SPARSE GOAL STEERING

SGS searches for a short motion-token prefix that leads a target agent toward a requested route. At selected decoding steps, it restricts the target’s available tokens. It scores the resulting joint prefix by continuing the simulation with the reference sampler. The simulator parameters and state-transition rule remain fixed.

2.1 PROBLEM SETTING

Let h_j denote the joint scene history at decoding step j , and let $\ell_{\theta,a}(h_j)$ be the frozen model’s logits for target agent a . Each motion token specifies a short transition that emits half a second of motion in our implementation. A reference sampler B maps these logits and a random vector $\xi_{a,j}$ to a token. This sampler includes any permanent vocabulary restriction, top- k truncation, and sampling temperature. Non-target agents always use B on the evolving joint history.

The goal g specifies a route and an oriented gate. We seek a prefix that increases the probability of reaching the gate before a deadline while satisfying target-level physical checks. Control is limited to an initial window and a maximum number of token replacements. The reference sampler generates all subsequent motion. We call a rollout generated entirely by B a native rollout. The replacement budget limits direct changes to the sampled tokens without constraining the distance from the native trajectory.

2.2 GOAL-DIRECTED DECODING

At a selected step, SGS previews each motion token from the target’s current state. Eligible tokens must belong to the reference vocabulary and satisfy local speed, acceleration, and yaw-rate limits. SGS ranks them by how closely their endpoints and headings follow the route. For token z , the cost is

$$c_g(z; h_j) = \|x_z - x^*\|_2^2 + 4 \text{ wrap}(\psi_z - \psi^*)^2, \quad (1)$$

where (x_z, ψ_z) is the previewed endpoint pose and (x^*, ψ^*) is a look-ahead pose on the requested route. Current speed determines the look-ahead distance. Appendix A gives its construction and the eligibility limits.

Let $S_w(h_j, g)$ contain up to the w lowest-cost eligible tokens. The model samples from this set using its original logits and temperature, applying top- k selection *within* S_w . Route geometry thus determines which tokens are available, while the model determines their probabilities. Restricting the support before top- k selection can make tokens outside the native top- k set available. If no token is eligible, SGS uses the reference token and marks the rollout as a failure under guarded success.

2.3 COUNTING INTERVENTIONS AND RELEASING CONTROL

A schedule $u = (u_0, \dots, u_{J-1})$ specifies the support width at each control step, with $u_j = 0$ requesting reference sampling. Our main configuration has four steps and widths in $\{0, 8, 32, 128\}$. An intervention counts toward the budget only if the mask changes the sampled token. We define the reference decision and replacement count as

$$z_{a,j}^{\text{ref}} = B(\ell_{\theta,a}(h_j^u), \xi_{a,j}), \quad K(u, \xi) = \sum_{j=0}^{J-1} \mathbf{1}[z_{a,j}^u \neq z_{a,j}^{\text{ref}}]. \quad (2)$$

Both decisions use the controlled history h_j^u and the same sampling noise. Their difference therefore measures the direct replacement at the current step, excluding effects of earlier interventions. Once K reaches the cap, all remaining requested masks are skipped.

The release time τ is the end of the last applied goal mask, or zero if none is applied. A mask can leave the sampled token unchanged, so release need not coincide with the last replacement. All agents then use B to continue from the changed history. Under fixed randomness and deterministic transitions, a rollout with no replacements matches its coupled native rollout (Appendix A.6). Differences that arise after release do not count toward the intervention budget. We measure these downstream effects separately.

2.4 PREFIX SEARCH AND EVALUATION

The main search space contains $4^4 = 256$ schedules, some of which yield identical joint prefixes. We simulate each candidate to a common two-second boundary and score it with two continuation probes. The score rewards guarded route success and penalizes goal error, physical violations, and token replacements. Its coefficients are fixed across methods (Appendix A). The native candidate is always included as an initial option.

We compare several search rules with the same control interface and scoring objective. Fixed-order search (SGS-fixed) and random search enumerate schedules without fitting a proposal distribution. A categorical cross-entropy method (CEM) uses high-scoring candidates to update a factorized distribution over the support width at each step. Beam search retains promising partial joint histories and scores completed prefixes with the same continuation objective. A repair heuristic modifies masks near the first step where the target strays from the route or violates a dynamics limit. When a candidate improves the best score so far and succeeds on both probes, the heuristic also tries removing masks.

Search is limited by the number of model predictions it uses. We reuse joint histories and cached probe outcomes to avoid repeating computations. Predictions used for rejected candidates and deletion trials still count toward the budget. After search, we fix the selected *joint* prefix through two seconds and evaluate it on 32 continuations with fresh sampling randomness. The probes used during search are excluded from this evaluation. Every method, including native sampling, uses the same two-second boundary. If control ends earlier, the rest of the selected prefix remains fixed across its evaluation continuations.

3 EXPERIMENTS

We evaluate whether sparse token replacements improve route success over native sample selection and whether this improvement persists after control ends. We also study the effects of search resources, changes to surrounding traffic, and control toward fixed-time endpoint goals.

3.1 EXPERIMENTAL SETUP

Simulator and tasks. We use a frozen SMART-family joint simulator with a 2,048-token motion vocabulary on Waymo scenes (Wu et al., 2024; Ettinger et al., 2021). The main study cohort contains 100 scenes and 217 branch requests, with one scene per detected group of overlapping source windows. Each request specifies an observed scene, a target agent, and a goal gate on a lane branch. The model predicts an eight-second joint future, and SGS may replace at most two target decisions in the first two seconds. Logged future motion is excluded from initialization and goal construction and is used only for overlap detection and the post-hoc labeling of alternative branches. Appendix A describes the grouping procedure and its limits.

Sampling references. We evaluate each method against both raw and filtered native sampling. The filtered reference permanently removes intrinsically high-acceleration vehicle tokens and remains active after goal control stops. Within each comparison, all methods use the same reference sampler. Sampler settings, checkpoint provenance, and filter details appear in Appendix A.

Success and uncertainty. Guarded route success requires a forward gate crossing by five seconds and valid target motion without detected collisions, offroad frames, or dynamics violations throughout the eight-second forecast. An empty support at an applied mask also counts as failure. These are target-level checks, with the full criteria given in Appendix A. Because arrival may precede release, Section 3.3 repeats the analysis under stricter timing conditions. We average continuations, search replicates, and tasks within each source group, then weight groups equally and estimate uncertainty with paired group bootstrapping. The 100-group comparison and the other diagnostic studies are development evidence. We report the prospectively reserved 512-group primary separately. Reservation from controller selection does not establish independent source logs or training-disjoint scenes, since source overlap screening is imperfect and checkpoint exposure remains unknown.

Comparators and search cost. We compare SGS with native sampling, best-of-16 native selection, and target-only reranking of nine native prefixes. Best-of-16 ranks complete native rollouts and retains the first two seconds of the selected rollout. We then evaluate this prefix on fresh continuations, as for SGS. Target-only reranking varies the target agent’s sampling noise during the prefix while keeping background noise fixed, and ranks candidates with the same probe objective. We cap search at 256 joint predictions. One joint prediction evaluates all agents in one rollout at one decoding step. Best-of-16 uses 256 predictions and target-only reranking uses 252. In the 100-group development comparison, we run three replicates and generate 32 fresh continuations from each selected prefix. These evaluation rollouts do not count toward the search budget.

Reserved primary confirmation. The primary confirmation evaluates the raw-reference SGS–CEM selected on a 64-group calibration set, using search seed 1717 and a 256-prediction cap. It compares CEM with native sampling and best-of-16 native selection on 512 reserved source groups. Confirmation outcomes were not used to select the controller. The primary estimates weight source groups equally and use paired source-group bootstrap intervals; Appendix B.9 gives the selection rule and audit details.

3.2 SPARSE STEERING IMPROVES ROUTE CONTROL

The primary confirmation tests whether calibration-selected SGS–CEM improves route success on the reserved cohort. Under raw sampling, CEM reaches 34.71% guarded success, compared with 23.92% for native sampling and 28.84% for best-of-16. The paired source-group gains are 10.79 percentage points (95% CI [9.26,12.36]) and 5.87 points (95% CI [4.51,7.27]), respectively. This result uses the prespecified raw-reference comparison and is separate from the development estimates below. Arm-level success and work appear in Table 8, and the same-model native-identity repair is detailed in Appendix B.9.

In the 100-group development comparison, SGS improves route success over both native sampling and native sample selection under the same search cap (Table 1 and Figure 2). Under the filtered reference, CEM raises success from 28.9% to 40.8%, gaining 12.0 percentage points over native sampling and 7.5 points over best-of-16. Improvements also hold under the raw reference. Thus

Table 1: Sparse steering and native selection. Scene-weighted guarded route success (%) over 100 source groups and 217 route tasks. SGS allows at most two replacements and a 256-prediction search cap. Replacements and search cost report filtered-reference means. Every selected prefix is evaluated on fresh continuations.

Method	Success (%)		Filtered mean	
	Raw	Filtered	Replacements	Search cost
Native	25.70	28.85	0.00	4.0
Best-of-16	30.27	33.31	0.00	256.0
Target-only reranking (9)	30.78	32.75	0.00	252.0
SGS-fixed	36.72	41.00	0.89	238.9
SGS-random	37.49	40.83	1.01	233.0
SGS-CEM	36.89	40.84	0.97	234.2
SGS-beam (8)	37.03	40.67	0.94	218.1
SGS-repair	36.79	40.64	0.96	236.8

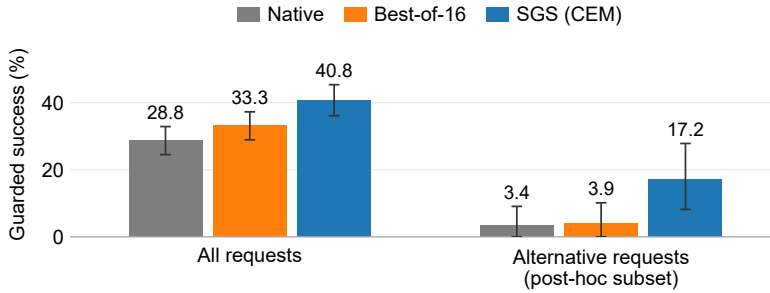


Figure 2: Sparse steering improves overall and alternative-route success. Bars show scene-weighted guarded success under the filtered reference for all 217 requests from 100 source groups and for a post-hoc subset of 48 alternative-branch requests from 43 groups. Alternative requests specify a branch different from the unique logged branch. Whiskers show 95% source-group bootstrap intervals.

the selected interventions produce prefixes with more successful fresh continuations than the tested native-selection baselines.

Within the development cohort, the improvement is shared by all five SGS search rules, whose filtered means differ by less than 0.4 percentage points. The paired comparisons do not identify a clear advantage for any one rule, suggesting that the observed development gain is not specific to CEM. We use CEM as a representative search rule below. Appendix B.1 gives exact paired estimates and intervals, and Table 3 includes the sparse-initialization and edit-quota variants.

Alternative branch requests. In a post-hoc analysis, we examine requests that depart from the unique logged branch. On this subset, native sampling and best-of-16 rarely reach the goal, whereas CEM achieves about 17% success (Figure 2). SGS can therefore induce some departures from recorded behavior, although most alternative requests remain unsuccessful. The main comparison retains all requests.

3.3 THE BENEFIT PERSISTS AFTER RELEASE

The primary criterion permits arrival during active control, so we reanalyze success after requiring first arrival strictly after the last applied mask. This post-hoc restriction changes filtered CEM success by less than one percentage point, with a similarly small change under raw sampling. Most successful controlled rollouts therefore reach the goal after guidance has ended.

The advantage also persists when first arrival must follow the common two-second prefix boundary, where fresh evaluation randomness begins (Figure 3). Filtered CEM retains a gain of about 12 percentage points over native sampling under this stricter condition. Both analyses count early arrivals as failures while keeping the original denominator and scene weighting, so the gain is not

produced by selecting only late-arriving worlds. Appendix A.7 reports the exact rates and paired intervals.

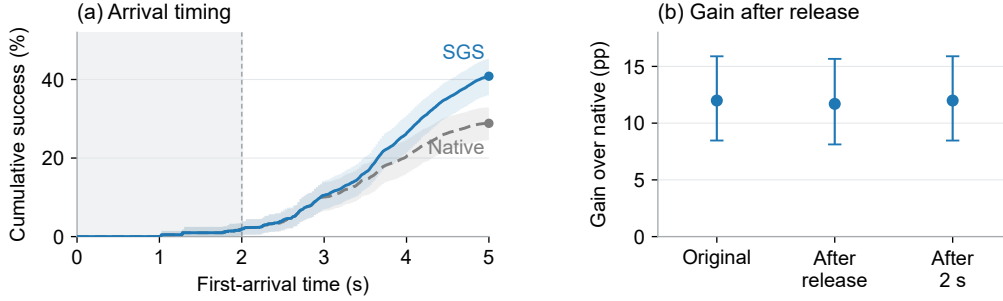


Figure 3: Route-control gains persist after release. Under the filtered reference, (a) shows cumulative guarded success by first-arrival time for SGS–CEM and native sampling. Panel (b) shows paired gains under the original criterion and when first arrival must follow each method’s release or the common 2 s prefix boundary. Early arrivals count as failures without changing the denominator or 5 s deadline. Gray shading marks the selected prefix. Bands and whiskers show 95% source-group bootstrap confidence intervals; bands in (a) are pointwise.

A sparse intervention can nevertheless have a substantial downstream effect. In a separate 20-group diagnostic, fixed-order SGS replaces approximately one decision directly but differs from its coupled native target sequence at about ten of sixteen forecast decisions. Much of the moving background remains close to native generation, with larger changes among successful rollouts with at least one replacement. The edit budget consequently describes the extent of direct control rather than the size of the final trajectory change. Appendix B.4 quantifies this propagation, and Appendix C shows selected examples.

3.4 CONTINUING THE TESTED GUIDANCE CAN REDUCE SUCCESS

To test whether guidance should continue, we compare releasing each selected SGS–CEM prefix with continuing width-32 route guidance through the goal deadline, using matched future randomness. A standalone controller applies the same rule from the forecast origin without search. This raw-reference comparison uses 20 groups drawn from the main study cohort, retaining all tasks and selections. Every arm shares the arrival deadline and full target checks, without requiring arrival after its own release time (Appendix D).

Release achieves about 35% success, compared with 24% under continued guidance from the identical prefix, while using about one replacement rather than five (Figure 4). The paired advantage remains positive after adjustment for all six comparisons. Standalone guidance also intervenes more often and reaches a lower success rate than release. Continuing this geometric rule can therefore reduce the benefit of a searched prefix despite adding goal-directed interventions. The result concerns this particular rule, rather than optimized sustained control. Appendix D gives exact estimates, paired intervals, and compute accounting.

The same 20-group study also tests whether the gain requires schedule search. A fixed width-32 mask at each of the four early clocks, with the same two-replacement cap but no schedule search, raises raw guarded success from 19.87% for its paired native arm to 26.32%, a gain of 6.45 percentage points with 1.80 actual replacements on average. All 20 groups pass their in-job metric and replay audits. This establishes a search-free sparse-intervention effect on the matched development cohort, while the searched CEM arm reaches 35.46% in the same diagnostic study. The fixed-mask result uses same-job native baselines because the older cross-study sustained-reference artifacts do not satisfy bitwise native identity for every group.

3.5 SEARCH RESOURCES AFFECT ROUTE SUCCESS

Additional search improves CEM in the separate 20-group resource study, whereas allowing more replacements does not consistently help (Figure 5). Raising the search cap from 256 to 1,024

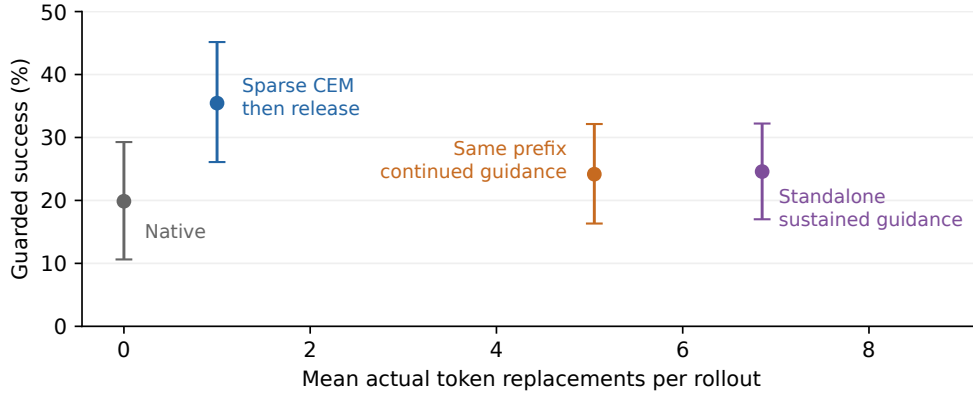


Figure 4: Releasing a searched prefix outperforms continued width-32 guidance. Points show guarded success and mean token replacements on 20 study groups containing 45 route requests under the raw reference. Sparse release and continued guidance share the selected 2 s prefix and continuation seeds; standalone guidance applies the same rule from the forecast origin without search. Vertical whiskers show marginal 95% source-group bootstrap intervals.

predictions adds about 3.5 percentage points of filtered success relative to the matched native baseline. At the default search cap, allowing four replacements instead of two leaves mean success slightly lower. In this comparison, exploring more candidate prefixes is more useful than increasing the number of permitted interventions. Appendix Figure 7 gives the resource curves and paired uncertainty.

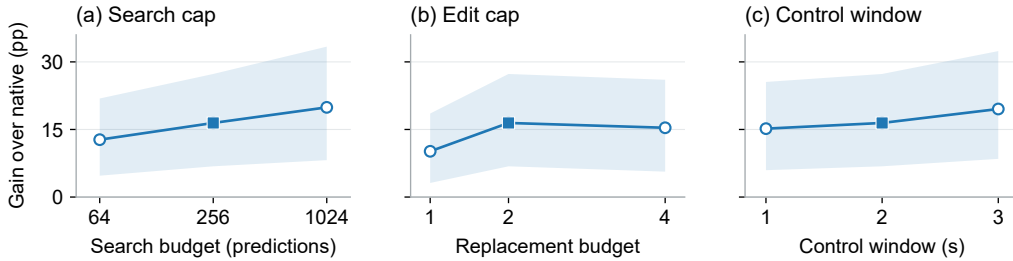


Figure 5: Sensitivity to search and intervention budgets. Each panel varies one setting from the default of 256 joint-world predictions, two permitted replacements, and a 2 s control window, holding the others fixed. Squares mark the default. Points show SGS-CEM’s guarded-success gain over matched native sampling on 20 study groups and 45 requests under the filtered reference. Bands show pointwise 95% source-group bootstrap confidence intervals, and lines connect evaluated settings. The native prefix boundary is matched to each control window.

The control window and number of probes also affect the search. A longer control window changes both the available interventions and the prefix from which evaluation starts. We therefore match the native reference to each window. More probes per candidate leave less compute for exploring other prefixes. In a separate scoring study, increasing the number of probes does not consistently improve success at fixed compute (Appendix B).

3.6 ARBITRARY ENDPOINT GOALS REMAIN DIFFICULT

Route-gate control does not readily extend to arbitrary fixed-time goals under the tested interface. With the same direct endpoint guidance, raw CEM succeeds on about 14% of route-derived targets but fewer than 1% of distance-matched arbitrary on-road targets. Unlike route-gate success, this task requires the target to occupy the endpoint at five seconds. The drop identifies a limitation of the tested controller and budget, while changes in goal geometry prevent attributing it solely to physical reachability (Appendix B).

4 RELATED WORK

Controllable multi-agent simulation. Trajeglish and SMART generate interactive traffic through discrete autoregressive motion models (Phillion et al., 2024; Wu et al., 2024). Several approaches learn interfaces for controlling generated behavior. ProSim conditions reactive agents on numerical, categorical, and textual prompts (Tan et al., 2025). ECoSim adds conditioning adapters, RLFTSim uses reinforcement-learning fine-tuning, and CNeVA learns behavior latents (Chen et al., 2026; Ahmadi et al., 2026; Lu et al., 2026). SGS instead restricts motion-token support during sampling. The controller requires no auxiliary training and keeps the simulator parameters fixed.

Inference-time guidance and search. PPLM guides a frozen language model through activation updates, and FUDGE reweights token probabilities with a learned future-attribute discriminator (Dathathri et al., 2020; Yang & Klein, 2021). In driving, Diffusion-ES combines a generative motion prior with black-box trajectory search (Yang et al., 2024). TOAD optimizes acceleration and yaw-rate sequences using Gaussian CEM, bicycle-model rollout, a frozen scorer, and planner-proposal initialization (Xu et al., 2026). Other methods guide generation by modifying the simulator’s decoding rule. CRAFT trains a contextual evaluator to reweight actions, SaFeR resamples tokens for feasibility-constrained adversarial generation, and Warm-K changes test-time sampling (Wang et al., 2026; Cui et al., 2026; Seong et al., 2025). SGS applies geometric support restrictions to a short motion-token prefix and limits the number of realized replacements.

Token intervention followed by completion. Token Steering injects action tokens derived from human guidance into an autoregressive vision–language–action policy and then resumes autoregressive completion (Chan & Kao, 2026). SGS also resumes generation after token intervention, but searches for goal-directed schedules in a reactive multi-agent simulator. It counts replacements at the controlled history and evaluates selected prefixes on fresh continuations. Our experiments compare search rules within this control interface. Our categorical CEM implementation searches SGS schedules. It does not reproduce TOAD.

5 DISCUSSION AND CONCLUSION

Sparse Goal Steering shows that a few early token replacements can redirect later motion in a frozen autoregressive traffic simulator. The development comparison finds gains across several search rules, and a prospectively reserved 512-group evaluation confirms that calibration-selected SGS–CEM outperforms native sampling and best-of-16 under the raw reference. Success also persists after guidance ends, while continuing the tested geometric rule reduces success from the same selected prefix. These findings support short interventions as a useful route-control interface. The resulting trajectory can differ substantially from native generation because the model continues from the altered history, and the poor arbitrary-endpoint results show that this interface does not provide general trajectory control.

The evidence comes from SMART-family checkpoints and study cohorts with imperfectly known source independence and training exposure. Our physical checks assess the target rather than joint-scene safety or realism. The fixed-mask result shows that schedule search is not required for a positive sparse-intervention effect on this cohort, but it does not establish that the fixed mask is the best search-free rule or that either controller transfers beyond the tested simulator, goals, and intervention rules. These limits also bound the result relative to external control systems.

AI USE STATEMENT

AI assistants were used for code and analysis, literature retrieval, figure production, and drafting and revising the manuscript.

ETHICS STATEMENT

The experiments concern generated multi-agent trajectories rather than deployment of a vehicle controller. Guarded success is an operational evaluation criterion, not evidence that generated scenes are safe for real-world execution. Manipulating simulated agents can produce misleading or hazardous scenarios if results are mistaken for validated real-world behavior. Dataset and checkpoint access must respect their licenses, and any released artifacts require an anonymity and privacy review.

REPRODUCIBILITY STATEMENT

The algorithm, sampler distinctions, evaluation endpoint, nested aggregation, and compute accounting are described in Sections 2 and 3 and Appendix A. Appendix B reports supplementary experiments. The accompanying materials include a machine-readable main-results aggregate and a table-verification script. Running the simulator also requires the original model, runtime configuration, scene assets, and dependencies. The supplementary package therefore does not provide a self-contained simulator.

REFERENCES

- Ehsan Ahmadi, Hunter Schofield, Behzad Khamidehi, Fazel Arasteh, Jinjun Shan, Lili Mou, Dongfeng Bai, and Kasra Rezaee. RLFTSim: Realistic and Controllable Multi-Agent Traffic Simulation via Reinforcement Learning Fine-Tuning. In *Proceedings of CVPR*, pp. 39734–39743, 2026. URL <https://arxiv.org/abs/2605.19033>.
- Jason Chan and Jonathan C. Kao. Steering Autoregressive Vision-Language-Action Policies via Action Token Intervention. arXiv:2606.15021, 2026. URL <https://arxiv.org/abs/2606.15021>.
- Yu-Hsiang Chen, Wei-Jer Chang, Yi-Ting Chen, and Masayoshi Tomizuka. ECoSim: Data Efficient Fine-Tuning for Controllable Traffic Simulation. arXiv:2607.00545, 2026. URL <https://arxiv.org/abs/2607.00545>.
- Jinlong Cui, Fenghua Liang, Guo Yang, Chengcheng Tang, and Jianxun Cui. SaFeR: Safety-Critical Scenario Generation for Autonomous Driving Test via Feasibility-Constrained Token Resampling. arXiv:2603.04071, 2026. URL <https://arxiv.org/abs/2603.04071>.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *International Conference on Learning Representations*, 2020. URL <https://arxiv.org/abs/1912.02164>.
- Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research*, 6:503–556, 2005. URL <https://www.jmlr.org/papers/v6/ernst05a.html>.
- Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset. In *Proceedings of ICCV*, pp. 9710–9719, 2021. URL <https://arxiv.org/abs/2104.10133>.
- Juanwu Lu, Junyu Zhu, and Ziran Wang. Controllable Sim Agents with Behavior Latents. arXiv:2607.02496, 2026. URL <https://arxiv.org/abs/2607.02496>.

- Jason D. Papastavrou, Srikanth Rajagopalan, and Anton J. Kleywegt. The Dynamic and Stochastic Knapsack Problem with Deadlines. *Management Science*, 42(12):1706–1718, 1996. doi: 10.1287/mnsc.42.12.1706.
- Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajenglish: Traffic Modeling as Next-Token Prediction. In *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=Z59Rb5bPPP>.
- Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient Cross-Entropy Method for Real-time Planning. In *Proceedings of the 2020 Conference on Robot Learning*, PMLR 155:1049–1065, 2021. URL <https://proceedings.mlr.press/v155/pinneri21a.html>.
- Hyunki Seong, Jeong-Kyun Lee, Heesoo Myeong, Yongho Shin, Hyun-Mook Cho, Duck Hoon Kim, Pranav Desai, and Monu Surana. Post-Training and Test-Time Scaling of Generative Agent Behavior Models for Interactive Autonomous Driving. arXiv:2512.13262, 2025. URL <https://arxiv.org/abs/2512.13262>.
- Shuhan Tan, Boris Ivanovic, Yuxiao Chen, Boyi Li, Xinshuo Weng, Yulong Cao, Philipp Krähenbühl, and Marco Pavone. ProSim: Promptable Closed-loop Traffic Simulation. In *Proceedings of the 2024 Conference on Robot Learning*, volume 270, pp. 5087–5105, 2025. URL <https://proceedings.mlr.press/v270/tan25a.html>.
- Ziyan Wang, Tan Xiang, Peng Chen, and Xintao Yan. Bridging Local Observation and Global Simulation in Closed-Loop Traffic Modeling. arXiv:2606.31844, 2026. URL <https://arxiv.org/abs/2606.31844>.
- Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. SMART: Scalable Multi-agent Real-time Motion Generation via Next-token Prediction. In *Advances in Neural Information Processing Systems*, 2024. URL <https://arxiv.org/abs/2405.15677>.
- Yihong Xu, Eloi Zablocki, Yuan Yin, Elias Ramzi, Ellington Kirby, Alexandre Boulch, and Matthieu Cord. Test-Time Trajectory Optimization for Autonomous Driving. arXiv:2606.07170, 2026. URL <https://arxiv.org/abs/2606.07170>.
- Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-ES: Gradient-free Planning with Diffusion for Autonomous and Instruction-guided Driving. In *Proceedings of CVPR*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Diffusion-ES_Gradient-free_Planning_with_Diffusion_for_Autonomous_and_Instruction-guided_Driving_CVPR_2024_paper.html.
- Kevin Yang and Dan Klein. FUDGE: Controlled Text Generation With Future Discriminators. In *Proceedings of NAACL-HLT*, pp. 3511–3535, 2021. URL <https://aclanthology.org/2021.naacl-main.276/>.

A IMPLEMENTATION AND REPRODUCIBILITY DETAILS

A.1 SIMULATOR AND INFERENCE CONFIGURATION

We use SMART’s environment with a local causal decoder wrapper and a frozen checkpoint in evaluation mode. Main-study inference uses FP32 with TF32 disabled, PyTorch 2.4.1, CUDA 12.1, and NVIDIA L40S hardware. Final continuations are evaluated in batches of 32. The forecast begins at source index 10. Valid target observations at indices 9 and 10 provide the initial position and velocity used to check dynamics at this boundary. We check source identities and manifest hashes. We also inspect a recorded check that changing logged future motion leaves the prepared model inputs unchanged. These checks do not rule out checkpoint training on evaluation scenes. The separate sustained-control study uses H100 hardware as specified in Appendix D.

The primary checkpoint is `CaTK_epoch10_pl.ckpt`, used with `train_local_catk.yaml`. Their SHA-256 digests and the decoder API match the experiment manifest. The checkpoint metadata contains no hyperparameter overrides. The corresponding SMART agent code uses `topk_prob`, $k = 5$, and temperature 1. The imported checks classify positive signed distance to the road as offroad and negative signed distance between agents as a collision. The source archive records these settings and checkpoint identities. Reproduction also requires the expanded configuration and dependency snapshot because the archive does not hash every imported module. A separate checkpoint audit verifies that all 811 tensors are preserved by the packaged primary checkpoint’s format conversion. Original pretraining, fine-tuning, model-selection, and vocabulary-construction exposure remain unresolved. Separation from controller development therefore does not establish that evaluation scenes were unseen by the backbone. The second checkpoint’s metadata also does not establish an independent training lineage.

A.2 SUPPORT CONSTRUCTION AND LOCAL GUARDS

We convert the requested lane sequence to a polyline, remove repeated adjacent points, and prepend a 100-metre extension along the initial route direction. We estimate speed v from the two most recent target centres, separated by 0.1-second intervals. At the forecast origin, both centres come from observations. We project the current centre onto the nearest polyline segment and advance along the route by $\text{clip}(0.5v, 1, 10)$ metres, capped at the route endpoint. The resulting position and segment heading define the look-ahead pose. We preview tokens to check continuity with observed motion and the three dynamics limits below. Among tokens in the reference vocabulary, those that pass these checks are stably sorted by Equation 1. The maximum retained width is 8, 32, or 128. In the filtered condition, a permanent intrinsic acceleration filter excludes 78 vehicle tokens whose centre second differences exceed 15 m/s^2 at a 0.1-second interval.

For a nonempty support S_w and reference-allowed vocabulary A_B , sampling is

$$\mathcal{T} = \text{TopK}_{z \in S_w} \ell_{\theta,a}(h_j)_z, \quad z_{a,j}^u = \arg \max_{z \in \mathcal{T}} \{\ell_{\theta,a}(h_j)_z / T_B + \xi_{a,j,z}\}, \quad (3)$$

where $\mathcal{T}$ contains the $\min(k, |S_w|)$ highest-logit tokens, T_B is the reference temperature, and ξ is keyed standard Gumbel noise. The reference token is computed at the same history with the same noise, replacing S_w by A_B . If the support is empty, we return the reference token and count the rollout as a guarded failure. Applied masks, sampled replacements, and empty-support events are recorded separately. Once the edit cap is reached, later requested masks are skipped.

A.3 SEARCH SCORE AND OPTIMIZERS

Candidates are ranked by

$$L(u) = 1000(1 - \bar{s}_{\text{probe}}) + \overline{\min(\ell_{\text{probe}}, 500)} + 0.01K(u), \quad (4)$$

where s is guarded success and ℓ is the per-probe surrogate. The main study uses search seeds 17, 117, and 217, probe seeds 100 and 101, and final evaluation seeds 1000–1031. We generate each noise entry from a key containing the scenario, seed, actor, clock, and token index. The vocabulary size determines the number of entries. The main study limits speed to 35 m/s, vector acceleration to 15 m/s^2 , and yaw rate to 0.8 rad/s. These checks include the observed-to-generated transition. The deadline condition is $\max(t_{\text{arrival}}, \tau) \leq 5$ seconds.

For successful probe worlds, the surrogate is $0.01K$. For unsuccessful worlds, its goal term is $\min_{t \in \{0.1, \dots, 5.0\}} (\|x_t - x_g\|_2 + 2|\text{wrap}(\psi_t - \psi_g)|)$, where (x_g, ψ_g) is the goal-gate pose. We evaluate distance and heading error at each frame before taking the minimum. The surrogate starts at 100 and adds a 100-point penalty for each of three conditions (failed target-feature checks, failed dynamics, and empty support). It also adds the number of offroad frames, twice the number of collision frames, and 0.05 times the maximum vector acceleration. Equation 4 caps this surrogate at 500 before averaging and adds the separate success and edit terms. The coefficients are fixed within each comparison. Increasing the probe count changes the relative weight of a one-probe success improvement. The objective therefore does not always rank a higher success rate ahead of a lower surrogate score.

Categorical search. The main CEM implementation maintains one distribution over $\{0, 8, 32, 128\}$ per decoding step, initially uniform. It evaluates native and dense width-32 schedules, then samples unseen schedules with population four and two elites. Given the old categorical probabilities p and elite frequencies e , it uses

$$p_{\text{new}} = 0.9(0.3p + 0.7e) + 0.1/4. \quad (5)$$

Sparse-initialized variants place probability 0.75 on native and $1/12$ on each other choice, with population four or eight and half-population elites. They do not require the same initial dense candidate. The allocation and calibration studies use earlier study results to select a population-16 configuration with native probability 0.25. The resource study holds population four, elite fraction 0.5, and native probability 0.25 fixed across all settings. All configurations optimize categorical schedules within SGS. They do not reproduce TOAD or continuous-control CEM variants (Xu et al., 2026; Pinneri et al., 2021).

Other search rules. Fixed ordering includes native, dense width-32, contiguous masks of widths 32/8/128, alternating patterns, and the remaining schedules. Late sweep evaluates the twelve single-mask schedules from the latest to the earliest decoding step before using the fixed order. Random search selects unseen aliases uniformly after native. Beam search retains eight deduplicated joint histories by a route/progress heuristic and scores completed prefixes with the same continuation objective. The edit-quota beam reserves 1, 3, and 4 positions for zero, one, and two actual changes, respectively, before filling remaining capacity.

Repair identifies the first prefix endpoint farther than three metres from the route or violating local dynamics, then proposes masks at nearby earlier or later steps with widths 32, 8, and 128. When no earlier departure is found, it defaults to the last step. Repair proposes masks based on departures within the prefix, without diagnosing failures later in the continuation. When a candidate improves the best score so far and both probes succeed, repair tries deleting active masks from late to early. Every trial consumes search budget.

A.4 CACHING, MODEL WORK, AND STOCHASTIC EVALUATION

A cached score is keyed by fixed context and the complete joint token-history array, including its shape. The target’s token trace alone would omit other actors’ histories. Probe caching additionally depends on the probe-seed tuple, and goal reward evaluation is recomputed for each goal. Capacity limits and alias collisions affect realized cost. We check the budget before paired probes, counting partial and discarded prefixes. After search, final evaluation verifies prefix geometry and compares the first batched continuation with a single-world replay.

All selected prefixes in the main experiment end at two seconds, even when the final applied mask occurred earlier. The search replicate determines randomness within the prefix. Fresh evaluation randomness begins at the common two-second boundary. Goal control can end before this boundary. Release-window studies change this common boundary and use matched native baselines within each cell.

Source grouping. Before selecting scenes, we screen source windows for matching moving-vehicle trajectory snippets and group linked windows into connected components. Links require matches from at least two actor pairs at three distinct start frames with a consistent temporal offset. We use a deterministic hash ordering to select one scene per detected component. The main cohort contains 100 such scenes, each with all requested branches retained. The overlap screen may inspect logged future

states, but these states do not enter goal construction or simulator initialization. Matched windows provide evidence of shared sources, but the screen may miss other overlaps. Bootstrap resampling uses the detected components and retains all nested tasks, search replicates, and continuation outcomes.

A.5 ENDPOINT SEMANTICS AND PHYSICAL SCOPE

The goal gate has a three-metre half-width and heading tolerance $\pi/6$. Gate crossing is detected when signed forward distance changes from negative to nonnegative. We interpolate crossing time, lateral displacement, and wrapped heading between the bracketing samples. Targets already beyond the gate do not count as crossing. The detector receives samples at 0.1–8.0 seconds, omitting the forecast origin. It therefore misses crossings before the first sample. We have not quantified this case’s frequency or effect.

All 80 future target states must be valid. Speed, vector acceleration, and yaw rate include the observed boundary. Target collision/offroad checks use Waymo feature functions and observed vehicle dimensions. These predicates do not check all-agent traffic rules or continuous-time collision freedom, so guarded success is not a safety certificate.

A.6 ZERO-CHANGE CONSISTENCY

Proposition (coupled reference consistency). Fix the initial context, reference sampler, deterministic state-transition implementation, and random keys for every actor and clock. If an SGS rollout has zero actual changes as defined in Equation 2, and subsequently uses the reference sampler, its entire joint token sequence and emitted trajectory equal those of the coupled native rollout.

Proof. The initial histories are equal. Suppose the histories agree at clock j . Every non-target actor uses the same reference sampling function with the same logits and random key in the two rollouts. For the target, either no mask is applied or zero-change accounting implies that its realized token equals the reference sample at the same history and key. The joint token decisions agree, and deterministic transitions yield equal next histories. Induction covers the intervention window and the reference-only continuation. $\square$

This result assumes fixed randomness. Selecting a zero-change sample based on a goal can still change the unconditional sampling distribution, and one or two direct changes can affect many subsequent predictions.

A.7 MAIN-COHORT FIRST-ARRIVAL AUDIT

We reanalyze saved gate-crossing times for all 15,624 selected records and 499,968 nested evaluation outcomes. We verify 3,906 duplicate native records and check that every task, method, reference, and seed is present. For each world, we require original guarded success and a first saved crossing satisfying either $t_{\text{first}} > \tau + 10^{-9}$ or $t_{\text{first}} > 2 + 10^{-9}$ seconds. Both rates retain all outcomes in the denominator. Native sampling has $\tau = 0$, whereas the common 2 s restriction gives every method the same temporal cutoff. The first criterion tests arrival after each method’s control ends. The second uses the same cutoff for every method.

Table 2: First-arrival sensitivity on the full main cohort. Scene-weighted guarded success (%). “After release” requires first crossing after the last applied mask. “After 2 s” requires first crossing after the common evaluation-prefix boundary. Original success remains the primary endpoint.

Method	Raw reference			Filtered reference		
	Original	After release	After 2 s	Original	After release	After 2 s
Native	25.70	25.70	24.11	28.85	28.85	26.87
Best-of-16	30.27	30.27	28.51	33.31	33.31	31.22
Target reranking	30.78	30.78	29.01	32.75	32.75	30.72
Late sweep	35.82	35.69	34.15	39.54	39.26	37.57
Sparse-init. CEM (4)	36.23	36.11	34.56	39.90	39.62	37.93
Sparse-init. CEM (8)	36.41	36.28	34.74	40.18	39.90	38.21
Quota beam	36.13	36.01	34.46	40.37	40.08	38.39
Repair	36.79	36.65	35.10	40.64	40.35	38.66
Beam (8)	37.03	36.90	35.36	40.67	40.38	38.70
Random	37.49	37.34	35.80	40.83	40.54	38.85
CEM	36.89	36.72	35.20	40.84	40.55	38.86
Fixed	36.72	36.55	35.03	41.00	40.70	39.02

Under the common first-arrival-after-2 s criterion, CEM gains 11.10 raw percentage points over native (95% paired source-component interval [7.98,14.57]) and 11.99 filtered points ([8.47,15.89]). Relative to best-of-16 under the same criterion, the gains are 6.69 raw points ([3.86,9.85]) and 7.63 filtered points ([4.57,11.07]). These exploratory intervals use the original 10,000-resample procedure.

For CEM, 33 raw and 56 filtered successful nested worlds first cross at or before their release time. These counts are not independent scenes. Before applying the timing restrictions, the analysis reproduces every original scene-weighted success mean exactly. Because the analysis uses archived gate events rather than recomputing crossings from trajectories, it does not resolve the forecast-origin edge case described above. The code and audit results accompany the manuscript source.

B SUPPLEMENTARY EXPERIMENTS

The supplementary studies examine search, resource allocation, propagation, and transfer using distinct study subsets or configurations. Each comparison uses its own matched native reference. The following subsections specify these settings separately from the 100-group main experiment.

B.1 MAIN-COHORT OPTIMIZER COMPARISON

For the main comparison, 95% intervals use 10,000 paired bootstrap resamples of the detected source groups. Filtered CEM gains 11.99 percentage points over native sampling ([8.47,15.89]) and 7.52 points over best-of-16 ([4.47,10.96]). The raw gains are 11.19 points ([8.08,14.64]) and 6.62 points ([3.78,9.82]), respectively.

The post-hoc alternative-branch subset contains 48 requests from 43 groups with a unique logged branch. Filtered success is 3.39% for native, 3.92% for best-of-16, 17.24% for CEM, and 18.24% for beam. These estimates describe departures from recorded behavior within the main study cohort.

Table 3: Comparison of schedule-search methods. Scene-weighted guarded route success (%) over 100 source groups and 217 route tasks. SGS allows at most two replacements under a 256-prediction search cap. Replacements and search cost report filtered-reference means. Native baselines make no goal-token replacements, and all methods use the same continuation procedure. Table 1 reports the primary comparisons.

Method	Success (%)		Filtered mean	
	Raw	Filtered	Replacements	Search cost
Native	25.70	28.85	0.00	4.0
Best-of-16	30.27	33.31	0.00	256.0
Target-only reranking (9)	30.78	32.75	0.00	252.0
SGS-late single-mask sweep	35.82	39.54	0.73	237.9
SGS-sparse-init. CEM (4)	36.23	39.90	0.92	234.8
SGS-sparse-init. CEM (8)	36.41	40.18	0.86	235.0
SGS-edit-quota beam	36.13	40.37	0.79	206.9
SGS-repair	36.79	40.64	0.96	236.8
SGS-beam (8)	37.03	40.67	0.94	218.1
SGS-random	37.49	40.83	1.01	233.0
SGS-CEM	36.89	40.84	0.97	234.2
SGS-fixed	36.72	41.00	0.89	238.9

B.2 SCHEDULE ENUMERATION AND PROBE ALLOCATION

We select 20 study groups by sorting source identifiers without using outcomes. These groups contain 45 requested branches. We use three search seeds and 32 final continuations. The exhaustive variants enumerate all 256 schedules, including those that yield identical prefixes. Four-probe enumeration improves over two-probe enumeration at substantially greater search cost (Table 4). All guarded successes in this study occur after release.

Table 4: Exhaustive schedule search. Success (%) and mean filtered search cost on a 20-group cohort separate from Table 1. The exhaustive variants use substantially more computation than the budgeted methods.

Method	Raw	Filtered	Filtered search cost
Native	18.03	19.87	4.0
SGS-fixed	33.57	38.65	239.4
SGS-CEM	34.88	37.86	231.9
SGS-repair	34.12	35.57	236.5
SGS-random	34.53	38.33	231.2
Finite-table DP, two probes	35.31	39.92	774.3
Enumerate, four probes	38.05	42.21	1519.1

Four-probe enumeration exceeds budgeted CEM by 4.35 filtered points (95% interval [0.52,9.32]), at approximately 6.6 times the logical work. Increasing exhaustive scoring from two to four probes adds 2.29 points ([0.81,4.01]). Both exhaustive searches exceed the main method’s compute cap.

The two-probe exhaustive result also has an exact finite-table dynamic-programming formulation. For a complete schedule prefix p , let $V_4(p) = L(p)$ with L from Equation 4, and set

$$V_j(p) = \min_{w \in \{0,8,32,128\}} V_{j+1}(p \parallel w), \quad j = 3, 2, 1, 0. \quad (6)$$

The minimizing choices recover the same selected schedules as exhaustive ranking in all 540 records across the two- and four-probe tables. For graph compression, a state includes the complete controlled history, actual replacement count, empty-support status, and last applied-mask step. Time and remaining replacements alone are insufficient because earlier tokens change the joint scene and continuation distribution. Merging states with matching saved joint histories and the same budget and validity metadata reduces the mean graph from 341 tree nodes to 78.2 states, with about 152 transitions, while preserving every terminal score and selection. These are offline table counts. The full set of terminal probe scores was still generated, so Equation 6 does not establish an online runtime saving or a 256-prediction implementation. The edit cap resembles an online knapsack capacity, but earlier actions alter the later scene and its possible rewards. The item-acceptance state used in stochastic knapsack work therefore does not suffice here (Papastavrou et al., 1996).

Fitted value policy. We also tested an approximate, adaptive Bellman policy trained from the audited four-probe finite table. At each of the four control steps, a stage-specific tree regressor estimates the terminal search loss for each requested width. We fit the last stage to observed terminal losses and each earlier stage to the minimum predicted value over actions at the observed successor state, as in fitted Q methods (Ernst et al., 2005). The state summary contains the remaining replacement budget, mask release and empty-support status, target speed, route progress and heading error, gate distance, and nearby-agent distances and counts. Five source-group folds exclude each tested group from training. The raw and filtered references have separate models.

At inference, the policy observes the joint scene after each chosen width and selects the next width using its fitted values. It uses no online continuation probes to choose a schedule. Unlike CEM’s open-loop schedule choice, this is an adaptive policy with an offline training cost. The label bank is the same four-probe enumeration whose generation required 426,918 logical predictions, and the broader archived diagnostic campaign required 814,626 actual predictions. We report online prediction heads separately from this training work. The selected joint prefix is evaluated on the same 32 fresh continuation seeds as the paired CEM and native runs. This comparison remains developmental because the 20 source groups and fitted model were used during method development.

Table 5: Cross-fitted value policy. Source-group-weighted guarded success (%) and mean online search heads on the 20-group enumeration cohort. The fitted policy’s offline label cost is described in the text.

Reference	Fitted Q	CEM	Native	Fitted heads	CEM heads
Raw	25.61	34.88	18.03	5.5	231.7
Filtered	26.46	37.86	19.87	5.7	231.9

Mean fitted-Q success exceeds native by 7.58 raw and 6.59 filtered percentage points, while falling below CEM by 9.27 raw points (95% paired source-group interval $[-15.29, -3.85]$) and 11.41 filtered points ($[-19.04, -4.92]$). Intervals use 10,000 paired source-group bootstrap resamples. The selected masks use 1.61 and 1.70 actual replacements on average. On this development cohort, the fitted value policy trades lower success for substantially fewer online predictions.

A separate fixed-budget study selects and freezes its CEM configuration based on two-probe earlier study results (population 16, native probability 0.25). With its separate seeds, native success is 17.88% raw and 20.00% filtered. The selected CEM configuration reaches 35.33% and 39.26%, fixed ordering 33.57% and 39.05%, and repair 34.15% and 35.83%. Repair minus CEM is -3.42 filtered points ($[-6.84, -0.62]$). Four rather than two probes reduces CEM success by 0.78 raw and 0.90 filtered points. Reserving 48 predictions to add two probes to two distinct finalists changes success by 0.00 raw and -0.16 filtered points. At this budget, evaluating more candidates was more useful than the tested increases in probe scoring. This comparison does not establish a general preference for noisier scores.

B.3 RESOURCE SENSITIVITY AND MATCHED BASELINES

Table 6 reports the filtered resource sweep, which evaluates all sampler and setting combinations in 840 experimental runs. The default allows two replacements by two seconds under a 256-prediction cap. Raising the CEM cap to 1,024 predictions adds 3.46 filtered percentage points relative to its matched native baseline, with a 95% paired group-bootstrap interval of $[0.52, 7.42]$. Work can fall below the cap when schedules are exhausted or prefixes are reused.

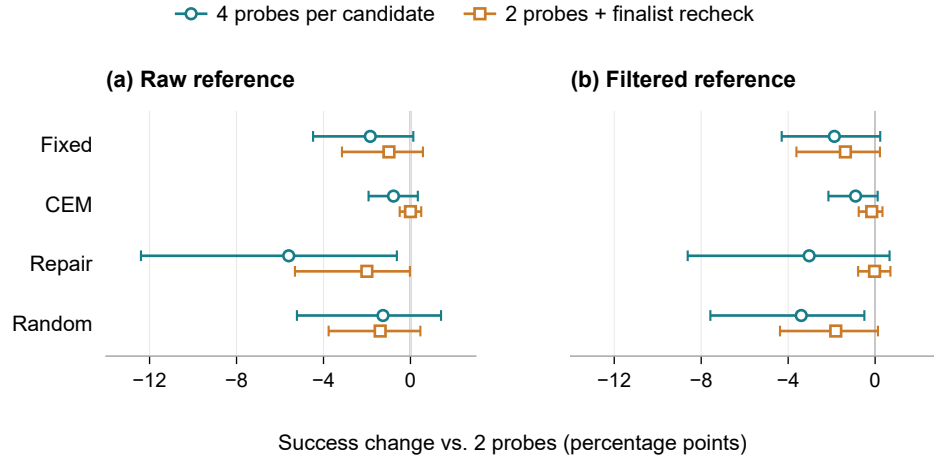


Figure 6: More continuation probes do not consistently improve fixed-budget search. Points show changes in guarded route success relative to two-probe scoring when using four probes per candidate or adding two probes to each of two finalists, under a 256-prediction search budget. Panels show raw and filtered references on 20 study groups and 45 requests. Whiskers show paired 95% source-group bootstrap intervals.

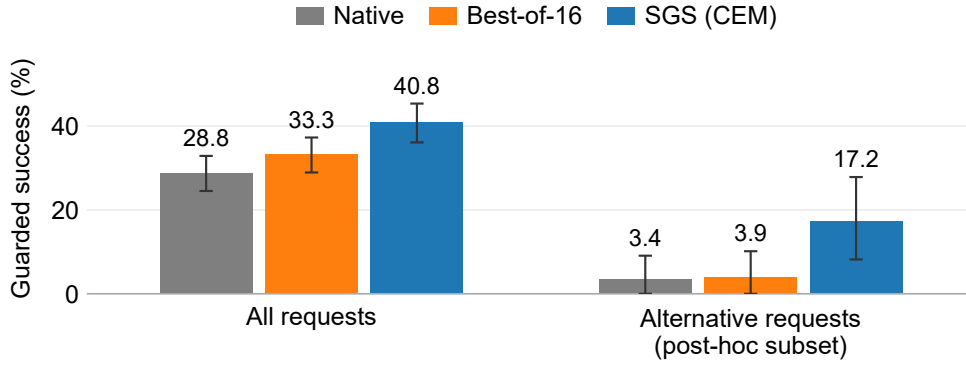


Figure 7: Resource sensitivity across search methods. Each row holds the optimizer fixed. Columns vary search budget, replacement budget, or control window around 256 joint-world predictions, two permitted replacements, and 2 s. Points show filtered success gain over the setting’s matched native reference, with 95% source-group bootstrap intervals. The study contains 20 groups and 45 requests. Release comparisons use setting-specific native controls because the continuation boundary changes. Lines connect evaluated settings.

Table 6: Resource sensitivity under the filtered reference. Scene-weighted success (%), with matched native references in each release-window cell.

Setting	Native	CEM	Fixed	Random	Repair
256 predictions, $K \leq 2$, 2 s (default)	23.07	39.52	37.17	38.95	36.90
64-prediction cap	23.07	35.81	35.68	32.17	24.97
1,024-prediction cap	23.07	42.98	43.14	42.57	42.54
One-change cap	23.07	33.23	32.53	32.81	32.06
Four-change cap	23.07	38.45	37.92	37.19	36.12
One-second window	23.46	38.65	37.02	36.65	36.95
Three-second window	22.86	42.41	39.17	40.29	35.16

A matched-baseline study contains 240 experimental runs. Raw/filtered success is 20.69/23.07% for native, 21.77/24.37% for best-of-16, 22.41/24.97% for target-only reranking, and 36.76/41.12% for width-eight beam. Beam averages 214.6/215.3 predictions. Its gains over native are 16.07 raw points ([5.70,27.32]) and 18.05 filtered points ([6.30,31.22]). CEM minus beam is -2.11 raw ($[-5.13,0.03]$)

and -1.60 filtered ($[-4.84, 1.16]$). Beam has the highest mean, although its paired intervals relative to CEM include zero.

B.4 PROPAGATION AND SUPPLEMENTARY PHYSICAL AUDIT

We classify background actors as moving when their native mean speed is at least 1 m/s. A change in validity counts as a failure to preserve an actor’s state. Across all 20 diagnostic groups, SGS-fixed preserves 86.83% raw and 86.42% filtered background tokens exactly, with 93.11% and 93.24% of frames within one metre. Among cases with an intervention and guarded success (12 groups raw, 11 filtered), these percentages are 77.81/77.99 for exact tokens and 88.12/89.04 for frame proximity. These subsets condition on intervention and success, so they describe a different population and do not estimate a controlled causal effect.

A separate swept-target audit on 120 diagnostic runs leaves fixed-order success unchanged at 33.57% raw and 38.65% filtered. CEM changes from 34.88 to 34.12% raw and from 37.86 to 37.42% filtered. Four-probe exhaustive search changes from 38.05 to 37.28% raw and 42.21 to 40.90% filtered. The all-agent road-clear check has a zero pass rate for filtered fixed ordering in this audit. Parked or offroad actors and map semantics contribute to failures, among other issues. These checks do not establish all-scene safety.

B.5 LEARNED SCHEDULE PROPOSAL

The learned extension ranks candidate schedules before the same online probe evaluation. It uses ridge regression with weighted feature standardization and regularization 10. The 189 features comprise nine scene features, 18 schedule features, and their 162 interactions. Scene features include normalized speed, approach distance, a constant-speed route-time estimate, goal displacement along/across heading, sine/cosine heading difference, cross-traffic count, and branch distance. Schedule features contain 12 width indicators and six pairwise clock-activity terms. Cross-fitting excludes the current source component. Before predicting on reserved calibration data, we check source-component membership.

The label bank contains 69,120 search-probe labels. Four-probe exhaustive label generation costs 426,918 logical predictions. Collecting the broader diagnostic bank, including other methods and evaluation, costs 6.26 worker-hours and 814,626 actual predictions. Fitting the ridge model takes 1.15 seconds, excluding data generation. The separately trained proposal model remains fixed online, and the base world model is unchanged. This learned extension is separate from the training-free core.

Table 7: Learned schedule proposals. Success (%) in a separate 20-group study with 120 experimental runs per budget.

Reference	Prediction cap	Learned	CEM	Fixed	Native
Raw	64	30.27	30.26	29.95	18.57
Filtered	64	36.07	35.43	35.53	22.08
Raw	256	38.67	36.72	32.94	18.57
Filtered	256	43.87	44.00	39.87	22.08

At 256 predictions, learned minus CEM is 1.95 raw points ($[0.26, 4.05]$) and -0.13 filtered ($[-1.82, 1.77]$). At 64 predictions the differences are 0.01 raw ($[-2.88, 2.45]$) and 0.64 filtered ($[-0.40, 1.85]$). Reducing the learned method’s own cap from 256 to 64 loses 8.39 raw points ($[-14.82, -3.24]$) and 7.80 filtered ($[-15.85, -1.56]$). Raw 64-prediction search latency is 2.17 seconds learned versus 1.63 seconds CEM in the recorded comparison. The learned proposal has mixed effects and loses success when its search cap is reduced fourfold.

B.6 FIXED-TIME ENDPOINT GOALS

The endpoint study contains 120 experimental runs. It requires the target to satisfy the endpoint condition at five seconds, so passing the location earlier does not count. Route-derived endpoints with route guidance, those same endpoints with direct guidance, and distance-matched arbitrary

on-road endpoints with direct guidance are distinct task conditions. For raw CEM their success rates are 18.75%, 13.57%, and 0.21%, while native rates are 0.70%, 0.70%, and 0.00%. The first CEM gain over native is 18.05 points ([8.36,28.75]). Arbitrary-goal success remains low. Raw random search reaches 2.34%, repair 0.21%, and fixed ordering 0.00%. Filtered CEM reaches 0.00% in that condition.

Only 15 of the 45 arbitrary goals, spanning 12 groups, pass a lane-graph connectivity check. Lane-graph connectivity does not establish dynamical reachability. Failure at the tested budget does not establish impossibility.

B.7 ROADWAY-DEFORMATION PILOT

We select 11 groups based on geometry for a pilot with 88 experimental runs, covering two references and four conditions (sham, far-field smooth deformation, left shift, right shift). We displace the road by one metre without closing roads or inserting barriers. Histories and goals remain fixed. Road edits cause native failures in only 0.71–2.13% of evaluated cases, leaving few failures on which to assess repair.

In the filtered left-shift condition, total success is 34.38% for CEM, 29.26% fixed, 28.27% repair, 30.82% random, and 18.61% native. Repaired edit-induced failures account for 1.14% of all cases for CEM/fixed and 0.57% for repair/random. No tested method repairs the right-shift-induced failures. Far-field deformation changes native behavior and is therefore not an inert negative control. This pilot does not establish reliable repair of roadway changes.

B.8 SECOND CHECKPOINT AND CALIBRATION

We repeat the raw-reference comparison with a second frozen SMART-family checkpoint in 60 experimental runs. All 811 tensors and 7,001,920 parameters load without mismatch, and replay checks pass without a fallback. Raw success is 11.88% native, 17.88% fixed, 17.84% repair, 19.71% CEM, and 20.34% random. Paired gains over native are 6.00 points ([1.94,10.79]), 5.95 ([1.83,10.77]), 7.83 ([2.84,14.22]), and 8.45 ([3.70,14.01]), respectively. This result replicates the comparison within the SMART family. We have not independently evaluated a second architecture.

A reserved calibration set contains 64 groups and 128 experimental runs. Raw native success is 26.75%, fixed 35.93%, repair 36.70%, tuned CEM 37.08%, and learned proposal 37.46%. The training-free selection rule chooses CEM, whose paired native gain is 10.33 points ([6.17,14.90]). Because these results selected CEM, they do not independently confirm its performance. We have not established whether all source groups are independent or whether the checkpoint was trained on these scenes.

B.9 PROSPECTIVE PRIMARY CONFIRMATION

The calibration rule selected among fixed, repair-prune, and tuned CEM using raw source-group mean success on the 64-group calibration set, breaking ties first by fewer actual replacements and then lexicographically by method name. We evaluated the selected CEM on all 512 reserved groups under the raw reference at search seed 1717 and a 256-prediction cap. The primary comparison uses native sampling and best-of-16 as baselines. Branches and continuations are averaged within each group, and groups receive equal weight.

SGS–CEM exceeds native sampling by 10.79 percentage points (paired 95% source-group bootstrap interval [9.26,12.36]) and best-of-16 by 5.87 points ([4.51,7.27]). For the two prespecified contrasts, the simultaneous 97.5% intervals are [9.07,12.57] and [4.32,7.49].

Identity and audit checks. The occurrence-aware identity correction for group 108 retained all four route requests and did not change controller settings or outcomes. After a separate audit detected native-identity mismatches, 13 full bundles from six groups were regenerated under the frozen manifest and same NVIDIA L40S model. Each repaired bundle passed metric, objective, replay, and paired-world audits. The strict native-identity check passed across all 1,152 primary bundles, and the original artifacts remain preserved. These checks do not resolve imperfect source independence, unknown checkpoint exposure, or the target-level scope of the physical gates.

Table 8: Primary 512-group confirmation. Raw guarded route success and mean realized search work. Confidence intervals for paired contrasts are reported in the text.

Arm	Success (%)	Mean replacements	Mean heads
Native	23.92	0.00	4.0
Best-of-16	28.84	0.00	256.0
SGS-CEM	34.71	1.06	233.7

C QUALITATIVE EXAMPLES

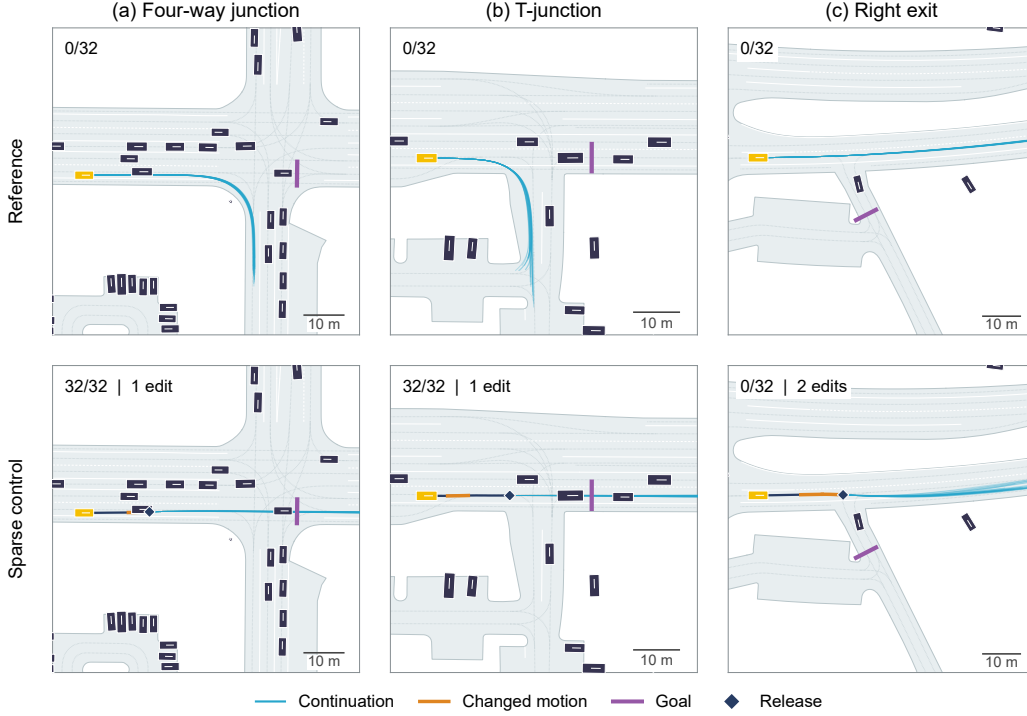


Figure 8: Examples of successful and unsuccessful sparse steering. Each column compares native sampling with SGS-repair under the filtered reference. All 32 target continuations are shown, and annotations report guarded-success counts and token replacements. Vehicle footprints mark initial background states. These selected examples come from an earlier eight-task study cohort, separate from the main evaluation.

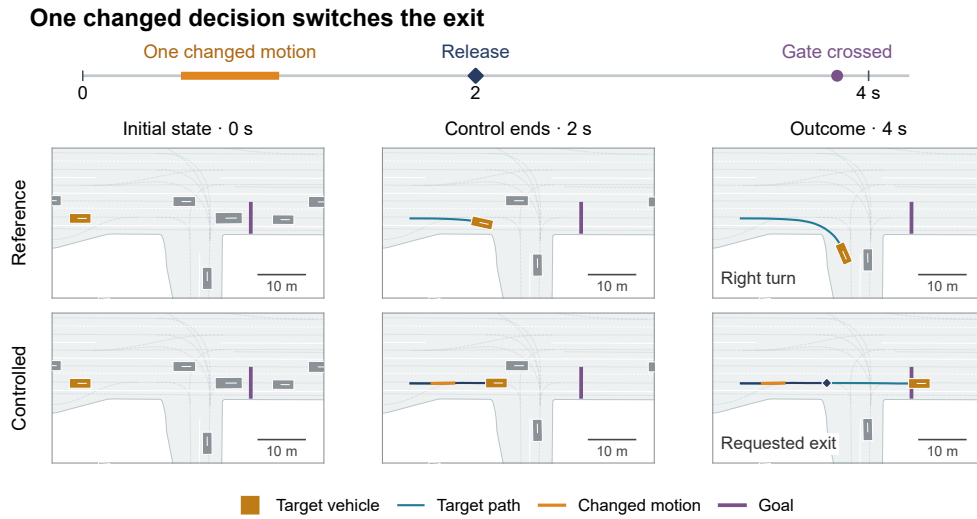


Figure 9: Paired scene snapshots before and after steering. One token replacement at 0.5–1.0 s redirects the target toward the requested exit. Guidance ends at 2 s, and the target first crosses the goal gate at 3.84 s. The panels show joint scene states at 0, 2, and 4 s from a common viewpoint. This selected example comes from the earlier eight-task study cohort.

D SPARSE RELEASE VERSUS SUSTAINED GUIDANCE

We compare sparse release and sustained guidance on 20 groups drawn from the main study cohort, containing 45 route requests and 32 paired continuations per arm and task. The CEM configuration has population 16, elite fraction 0.25, native probability 0.25, a 256-prediction search cap, two search probes, and at most two actual replacements over four early decision clocks. Search uses seed 617 and evaluation uses seeds 9000–9031. This shared development population makes the comparison a diagnostic rather than an independent replication.

The release and continued-guidance arms share the selected joint prefix and continuation keys. Continued guidance applies width-32 route support until five seconds. Standalone guidance applies that rule from time zero and performs no schedule search. Its additional interventions are not subject to the sparse two-edit cap. Failed and zero-edit selections remain in every comparison. All arms use the same goal deadline and full eight-second target assessment.

Table 9: Sparse release versus continued guidance. Intervals are marginal 95% source-group bootstrap intervals. Replacements count same-history token changes.

Arm	Success (%)	95% interval	Mean replacements
Native	19.87	[10.62,29.27]	0.000
Sparse CEM, then release	35.46	[26.09,45.16]	1.000
Same prefix, continued guidance	24.17	[16.33,32.14]	5.051
Standalone sustained guidance	24.58	[17.01,32.21]	6.853

We average tasks and continuations within source groups and use 20,000 paired group-bootstrap draws. Across the six pairwise contrasts, the release advantage over continued guidance is 11.29 percentage points, with a marginal interval [4.10,19.84] and a Bonferroni-adjusted interval [2.15,23.01]. Against standalone sustained guidance, the advantage is 10.87 points, with intervals [3.42,19.65] and [1.31,23.12], respectively. All six paired contrasts are retained in the accompanying numerical results, including those whose intervals contain zero.

Continuing guidance also increases the fraction of worlds with a detected target collision from 0.12% to 18.75%, and the fraction with target offroad frames from 7.62% to 32.01%, relative to release from the same prefix. These failure categories overlap and do not provide an additive or complete causal explanation of the success difference. They concern the target, not non-target safety.

Independent target-metric recomputation covers all evaluation tails. Exact reference-decoder replay covers the first tail per controlled arm and task, complemented by single-versus-batch equality checks. These checks do not establish all-agent safety. The study uses H100 hardware with FP32 inference and 32-world continuation batches. Its aggregate recorded model work is 72,757 logical predictions across search, probes, evaluations, and parity checks, excluding independent audits. CEM search is shared between the matched-prefix arms. Standalone guidance uses less search, so this is not an equal-compute comparison. Geometry previews and independent audits incur additional work. Per-arm wall time was not separately instrumented.

The comparison isolates whether to continue one geometric guidance rule after a searched prefix. Its result is specific to that rule and does not establish an ordering between release and optimized sustained control.

D.1 SEARCH-FREE PAIRED-NATIVE REPAIR

The same 20-group diagnostic cohort includes a search-free fixed-mask arm. The controller applies width-32 support at each of four half-second clocks, stops after two actual replacements, and performs no schedule search or candidate selection. Each group has a native arm generated by the same fixed-mask job, followed by 32 continuation seeds per task. All 20 groups have passing in-job metric and replay audits.

The exploratory marginal 95% source-group bootstrap interval for the fixed-minus-native difference is $[-0.63, 13.98]$ percentage points. The result is a valid paired-native repair study. It does not claim the original cross-study identity gate because the older sustained-reference artifacts still fail bitwise

Table 10: Search-free sparse intervention with paired-native baselines. Raw guarded route success is averaged within groups before equal group weighting. The interval is an exploratory marginal bootstrap interval over the 20 group-level fixed-minus-native differences.

Arm	Success (%)	Mean replacements	Groups	Audits passed
Native, same job	19.87	0.00	20	20/20
Fixed width-32 mask	26.32	1.80	20	20/20
Difference	+6.45	1.80	20	—

native identity for some groups. The aggregate and its audit receipts are included in the evidence package.