WHERE DOES OCCUPANCY LATENCY GO? A PER-STAGE DECOMPOSITION OF 3D OCCUPANCY INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluation of 3D semantic occupancy prediction relies almost exclusively on accuracy metrics, namely mean intersection over union (mIoU) and its ray-based variants, measured on server-grade GPUs. Deployment-related metrics do appear in the literature, but they are collected on an ad hoc basis across different hardware platforms under unspecified settings. Among the works we surveyed, none reports these metrics as a function of computational budget. To address this gap, we provide the measurements needed for budget-aware evaluation by decomposing occupancy inference latency stage by stage. We instrument six public configurations and attribute latency to six concrete stages, repeat runs under a common fixed protocol, report median and 95th-percentile (p95) latency, and account for the residual time outside the measured stages. Extensive experiments show that the 2D image backbone, rather than the occupancy head, dominates single-frame latency. A two-frame temporal window shifts the dominant cost to view transformation, while scaling up the backbone makes image feature extraction a co-dominant cost again. Comparisons among variants within the same architecture lineage further show that the differences are concentrated primarily in the bird's-eye-view (BEV) encoder backbone and occupancy head, where they correspond to approximately constant absolute latency savings. We use this decomposition to introduce a new reporting convention that expresses accuracy as an explicit function of deployment budget. We apply it to Occ3D-nuScenes and make the measurement framework and results publicly available.

1 INTRODUCTION

Camera-based 3D semantic occupancy prediction represents a scene by assigning semantic labels to voxels in an ego-centric volume. Benchmarks such as Occ3D-nuScenes (Tian et al., 2023) and metrics such as RayIoU (Liu et al., 2024) support comparisons of prediction quality. Selecting a model for a perception system also requires an estimate of inference cost. On a fixed platform, end-to-end latency determines which configurations meet a given budget, while stage-level measurements identify where optimisation could be most useful. Together, these measurements support both model selection and diagnosis.

Existing occupancy methods already examine efficiency at several levels. FlashOCC reports the cost of the BEV encoder and occupancy head separately from the remaining computation (Yu et al., 2023), while FastOcc reports a three-part decomposition into 2D, 2D-to-3D and 3D computation across model and runtime settings (Hou et al., 2024). These analyses show the value of decomposing inference cost. A finer breakdown can reveal how individual stages, time outside the measured stages, and variation across runs contribute to the end-to-end difference between configurations in a model family.

A percentage speedup alone cannot resolve these contributions. Its magnitude depends on both the absolute time difference and the total latency used as the denominator. Grouping several components can likewise conceal an increase in one component behind a reduction in another. Interpreting these differences requires explicit stage boundaries, accumulation over repeated calls, and separate reporting of the residual between end-to-end latency and the measured stage times.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this work, we conduct a per-stage analysis of six public configurations from the BEVDet-OCC and FlashOCC lineage on Occ3D-nuScenes. We time six named architectural stages and report the residual separately. We then examine how differences in stage costs relate to the end-to-end difference and its variation across runs. Our main contributions are as follows:

- **A per-stage measurement protocol.** We measure the image backbone, image neck, view transformation, BEV encoder backbone, BEV encoder neck and occupancy head, accumulating repeated module invocations within each forward pass. We report median and p95 latency and the residual, and assess run-to-run variation and sensitivity to stage-boundary synchronisation.
- **An empirical analysis of latency composition.** We compare the measured stages of two occupancy model variants at ResNet-50 and Swin-Base scales, complemented by two single-frame FlashOCC configurations. The analysis separates differences in individual stages from the end-to-end difference and examines which observed differences persist across repeated runs.
- **Model selection guided by end-to-end measurements.** We report the highest measured accuracy among configurations whose median latency meets a given budget. Making the latency budget explicit connects the stage analysis to the accuracy–latency trade-off within the evaluated configurations and measurement setting, supporting model selection.

2 Related Work

3D occupancy prediction. Camera-based occupancy methods use several scene representations. Some lift 2D features into a 3D volume through an explicit view transformation (Philion & Fidler, 2020) and predict voxel semantics in that volume (Sima et al., 2023; Wei et al., 2023). Others decode a BEV representation densely into voxels (Huang et al., 2021; Li et al., 2023), represent the scene on three orthogonal planes (Huang et al., 2023), or use sparse queries (Liu et al., 2024). These approaches share a voxel-wise prediction target but introduce the third dimension at different points. Consequently, the computational load falls on different components: a volumetric head (Sima et al., 2023; Wei et al., 2023), a plane-based decoder (Huang et al., 2023), or a dense BEV decoder (Li et al., 2023).

The evaluated temporal and stereo configurations build on BEVDet4D and BEVStereo (Huang & Huang, 2022; Li et al., 2022). Efficiency-oriented occupancy methods change different parts of this computation. FlashOCC uses 2D operators in its BEV encoder and occupancy head (Yu et al., 2023), while FastOcc fuses 2D and 3D branches (Hou et al., 2024). Fully Sparse Occupancy uses sparse queries (Liu et al., 2024), and PanoOcc unifies occupancy and panoptic segmentation (Wang et al., 2024). Their reported runtime settings are summarised in Table 14. FlashOCC's grouped BEV-encoder/head analysis and FastOcc's three-stage runtime breakdown already demonstrate the value of component-level measurement. Our six-stage analysis further separates backbone, neck and head costs, reports the residual explicitly, and compares local differences with end-to-end variation across runs.

Efficiency and deployment in 3D perception. Occupancy runtime has also been studied on embedded hardware. OneOcc reports four runtime stages on a Jetson AGX Orin for panoramic occupancy prediction in legged robotics (Shi et al., 2026); CMOP reports deployment alongside SurroundOcc on the same device family (Xu et al., 2025). These studies extend runtime analysis to embedded occupancy systems, with tasks and execution settings that differ from our Occ3D-nuScenes FP32 evaluation. Such differences matter when interpreting timings across papers, as reflected in an occupancy survey that aggregates results from heterogeneous settings (Zhang et al., 2024). For 3D detection, latency-aware streaming evaluation incorporates runtime into an online perception benchmark (Peng et al., 2025). Our median-budget example focuses on selecting among occupancy configurations measured on a fixed platform in a given session.

Evaluation validity and budgeted selection. RayIoU addresses limitations of voxel-wise accuracy evaluation (Liu et al., 2024), while UniOcc studies occupancy evaluation and label-free metrics (Wang et al., 2025). Runtime analysis adds computational cost to this assessment of prediction quality. Accuracy–latency trade-offs are also central to hardware-aware architecture search (Lu et al., 2021), and hardware benchmarks make execution settings explicit (Runge et al., 2020). We use this

established perspective to report accuracy conditional on a measured median-latency budget. End-to-end latency determines admission under this rule, while stage decomposition helps interpret the measured cost.

3 METHOD

3.1 STAGE TAXONOMY

We map each configuration to six module-level stages and accumulate their execution times within each forward pass. Using the same stage taxonomy across configurations provides a common architectural basis for comparing their profiles with end-to-end latency:

#	Stage	Role
S1	image backbone	2D feature extraction (ResNet / Swin)
S2	image neck	2D feature fusion (FPN)
S3	view transformation	2D $\rightarrow$ BEV/3D lifting (LSS, optionally stereo)
S4	BEV encoder backbone	BEV-space feature extraction
S5	BEV encoder neck	BEV-space feature fusion
S6	occupancy head	per-voxel semantic prediction

Stages S1–S2 form the *2D image encoder*: the backbone extracts image features and the neck fuses them. S3 lifts these features into a BEV or 3D representation. S4–S5 then extract and fuse BEV-side features, and S6 predicts voxel semantics. For multi-frame configurations, a named stage can be called repeatedly within one forward pass; its reported cost includes all recorded invocations. Table 6 lists these invocation counts.

The taxonomy defines all possible module regions covered by the timers. Initial input transfer to the device occurs before timing and is excluded from both end-to-end latency and the residual. Input preparation and output processing within the timed model call are included in end-to-end latency, with work outside the six stages contributing to the residual. Such work may include auxiliary learned operations, paths that bypass the instrumented module boundaries, and framework or pre-/post-processing operations. Invocation counts reveal repeated module calls.

3.2 MEASUREMENT PROTOCOL

We use `time.perf_counter` for wall-clock timing with explicit device synchronisation. All configurations are measured on a single NVIDIA L40 (46 GB), in PyTorch FP32, with the device otherwise idle. For each configuration, the measurement proceeds as follows:

1. **Prepare and warm up.** Set the model to evaluation mode, load the first batch from the non-shuffled validation loader at batch size 1, and transfer it to the device before timing. Reuse this input and discard 10 warm-up iterations to reduce initialisation effects such as lazy kernel compilation and cuDNN autotuning. Reset the peak-memory counter after warm-up.
2. **Measure 50 forward passes.** Call `torch.cuda.synchronize()` before and after each timed region. Synchronisation waits for device work to finish. The timer therefore captures elapsed wall-clock time, including host-side overhead within the region, rather than just kernel-enqueue time.
3. **Accumulate within each pass.** Sum the elapsed times of every recorded invocation of a stage before computing statistics across passes. Multi-frame models invoke the image encoder and view transformation repeatedly, so retaining only the last call would omit earlier work.
4. **Summarise the run.** Report median and p95 latency for each stage and for the end-to-end path, together with the residual. Read the peak allocated GPU memory after the timed run and report it in decimal MB (allocated bytes divided by 10^6).

The end-to-end timer encloses the model call and waits for device completion before stopping. In addition to the six stage hooks, the main harness times `prepare_inputs`, `simple_test` and `simple_test_occ`, with synchronisation before and after each call. These regions enclose stage

calls and support the residual breakdown in Table 7. Their instrumentation overhead is included in the end-to-end measurement.

Timing notation and aggregation. Let c index a configuration, k a timed forward pass, and $s \in \{1, \ldots, 6\}$ a stage. If $d_{c,s,k,j}$ is the elapsed time of its jth recorded invocation, its per-pass cost is $t_{c,s,k} = \sum_j d_{c,s,k,j}$. Let $T_{c,k}$ denote end-to-end latency for that pass. Stage medians and p95 values are computed from the accumulated $t_{c,s,k}$, not from individual invocations. The end-to-end median $\ell_c = \mathrm{median}_k\, T_{c,k}$ is the statistic used for budget admission below.

The median is computed over the 50 measurements. The harness computes p95 by sorting these measurements and selecting zero-based index $\lfloor 0.95 \times 50 \rfloor = 47$, the 48th ordered value, without interpolation.

The main harness reports the residual as $R_c = \ell_c - \sum_{s=1}^{6} \mathrm{median}_k\, t_{c,s,k}$, rounding the total and individual stage medians to three decimal places in milliseconds before subtraction. he per-pass residual $r_{c,k} = T_{c,k} - \sum_{s=1}^{6} t_{c,s,k}$ is a distinct quantity: its median generally differs from R_c and is not the summary computed by the main harness. Table 7 uses the difference-of-medians convention for its total residual. Its component rows are medians of per-pass measurements or derived remainders, so they need not add to R_c.

Under the fixed-input protocol, p95 characterises variation across repeated executions of the same input. Run-to-run variation is reported in Table 11; the separate distinct-frame experiment examines sensitivity to changing inputs. Neither run-to-run nor distinct-input measurements alone establish a deployment deadline guarantee.All reported configurations therefore use the wall-clock protocol.

The synchronisation ablation (Table 12) compares stage hooks with and without synchronisation. Both modes omit the three enclosing function wrappers used by the main harness, making the ablation a distinct measurement setup. Their end-to-end difference quantifies sensitivity to stage-boundary synchronisation in that setup. It neither validates individual stage times nor bounds all sources of instrumentation overhead. Absolute timings and stage proportions are specific to the evaluated hardware and runtime.

The budget-explicit protocol. We use end-to-end latency for model selection and the stage profile for diagnosis. For a fixed platform, runtime and measurement session, the budget-explicit protocol is:

> Given a platform and a latency budget B, the admissible set contains configurations whose measured median latency is at most B. For each budget, a budget-explicit evaluation reports the highest accuracy among admissible configurations, stating the budget alongside the accuracy.

Here, admission means $\ell_c \leq B$ and is restricted to the evaluated configurations. Accuracy is evaluated separately from the repeated-input timing run. The p95 is reported alongside the median but does not determine admission under this rule. The illustrative budgets are round numbers above the measured medians of the configurations admitted at each tier. They are not sensor-period deadlines. A deadline-constrained evaluation would additionally need a specified miss-rate tolerance and representative input sequences. Peak memory is reported as a separate feasibility constraint.

Admission is conditional on the measurement session: configurations near a budget boundary can change status when their measured median changes. Table 11 reports this variability, including the larger spread at Swin-Base scale. The resulting accuracy–latency comparison is conditional on the chosen statistic, budget, configurations and runtime.

3.3 Models and data

All models are evaluated on **Occ3D-nuScenes** (Tian et al., 2023) with the standard 18-class label set (17 nuScenes-lidarseg classes plus free space), a 0.4 m voxel grid over a $[-40, 40] \times [-40, 40] \times [-1, 5.4]$ m volume, and the visibility-masked mIoU convention in which voxels unobserved in the camera view are excluded.

We study two variants from one architecture lineage:

- **BEVDet-OCC** (Huang et al., 2021), the dense baseline.
- **FlashOCC** (Yu et al., 2023), a variant that replaces the BEV encoder and the occupancy head with a lighter design.

Both variants are evaluated in the two-frame ("4D-stereo") setting at two scales: ResNet-50 with 256×704 inputs and Swin-Base with 512×1408 inputs. Temporal fusion follows BEVDet4D (Huang & Huang, 2022), and stereo depth estimation follows BEVStereo (Li et al., 2022). FlashOCC is additionally evaluated in two single-frame configurations, M1 (256 BEV-side channels) and M0 (128 channels, with a coarser depth resolution). Appendix A lists all six measured configurations. Single-frame BEVDet-OCC is outside this measured set.

Scope of the comparisons. At each scale, the two-frame variants use the same backbone family and nominal input resolution. Their S1–S3 settings, weights and execution paths have not been verified as identical. Differences in S4–S6 therefore describe the local costs of the evaluated configurations, without isolating a component substitution that explains the entire speedup. End-to-end differences also depend on S1–S3 and the residual. Likewise, the single-frame versus two-frame comparison couples temporal processing with stereo estimation, the ResNet-50 versus Swin-Base comparison changes both backbone and input resolution, and M0 versus M1 changes both width and depth resolution. The comparisons reveal how latency is distributed across these configurations, but the coupled changes prevent attribution to any single factor.

4 Experiments

4.1 Setup

We evaluate the six configurations defined in §3 on a single NVIDIA L40 (46 GB), using driver 580.178.04 and PyTorch 1.10 (CUDA 11.1), in FP32 with the device otherwise idle. Accuracy uses the Occ3D-nuScenes validation split (6,019 frames) and the visibility-masked mIoU convention. Timing uses one fixed validation input repeated 50 times after 10 warm-up iterations, with batch size 1 and the synchronised wall-clock protocol of §3. Accuracy therefore covers the validation split, whereas the main timing results characterise repeated execution of one input. We analyse the distinct-input measurements separately below.

The configuration and released-checkpoint identifiers are listed in the supplementary package. Our accuracy values differ from the projects' recorded references by at most 0.43 mIoU points (Table 9). For FlashOCC Swin-Base, the reference comes from a sibling configuration with different learning-rate and loss settings.

4.2 Where occupancy latency goes: single frame

Figure 1 and Table 8 show that **the image backbone is the largest measured stage in both single-frame FlashOCC configurations**. It accounts for 43.0% of end-to-end median latency in M1 and 53.6% in M0, while the occupancy head accounts for 7.1% and 3.6%, respectively. The backbone times are nearly equal at 10.69 and 10.70 ms, so the difference in their shares mainly reflects total latency. View transformation contributes about 11%, and the full BEV side (S4–S6) contributes 36.8% in M1 and 24.5% in M0. Tables 3 and 4 give the stage statistics. Each share uses the stage median divided by the end-to-end median.

The small head shares indicate that reducing head latency alone would affect only a small fraction of total inference time in these models. Within the BEV side, the largest component is the neck in M1 but the backbone in M0 (Table 8). Both configurations use FlashOCC's lighter BEV-side design. The corresponding head share in single-frame BEVDet-OCC remains unmeasured.

The single-frame residual is 1.60–1.61 ms, or 6.5–8.0% of end-to-end latency. Post-processing is its largest reported component, at 1.05–1.08 ms (Table 7). This component is computed by subtracting head time from the enclosing output routine, so it does not isolate argmax. The inter-stage remainder is 0.25–0.27 ms in these profiles and 7.0–87.9 ms in the multi-frame profiles.

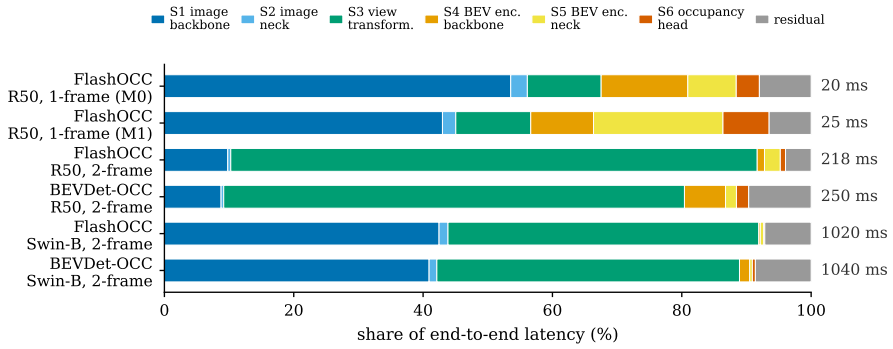

Figure 1: Latency composition across the six configurations, using stage medians as shares of the end-to-end median. The image backbone is the largest stage in the single-frame FlashOCC profiles; view transformation dominates the two-frame R50 stereo profiles; the two stages are co-dominant at Swin-Base scale. Temporal processing and stereo estimation change together, as do backbone and input resolution. The residual follows the difference-of-medians convention in §3.

4.3 Stage differences between model variants

We next compare BEVDet-OCC with FlashOCC in the two-frame stereo setting at each backbone scale (Tables 1 and 5). Figure 2 shows the stage medians, and Figure 3 their signed differences. Throughout these comparisons, Δ denotes BEVDet-OCC minus FlashOCC: a positive value means lower cost for FlashOCC. Each pair shares a backbone family and nominal input resolution, with the remaining comparison constraints described in §3.

Stage	BEVDet-OCC (median, ms)	FlashOCC (median, ms)	Δ
S1 image backbone	21.95	21.39	+0.56
S2 image neck	1.03	1.02	+0.01
S3 view transformation	178.40	177.71	+0.70
S4 BEV encoder backbone	**15.90**	**2.46**	**+13.45**
S5 BEV encoder neck	4.23	5.34	−1.11
S6 occupancy head	**4.74**	**1.76**	**+2.97**
sum of stages	*226.25*	*209.67*	*+16.58*
residual	*24.18*	*8.64*	*+15.54*
end-to-end	**250.43**	**218.31**	**+32.11**
peak memory (MB)	3736	1965	+1771

Table 1: Two-frame R50 configuration comparison at 256×704 input. All differences use BEVDet-OCC minus FlashOCC, including memory. Rounding can produce a 0.01 ms discrepancy when subtracting displayed medians. The recorded calls per forward pass are two for S1–S3 and one for S4–S6. Shared stage names do not establish identical settings or weights.

At R50 scale, the reported end-to-end difference is 32.11 ms. Of this difference, 16.42 ms lies in S4 and S6: 13.45 ms in the BEV encoder backbone and 2.97 ms in the head. FlashOCC's neck takes 1.11 ms longer in S5, offsetting part of this reduction. Consequently, the *net BEV-side difference* across S4–S6 is 15.31 ms, computed from the reported deltas. S1–S3 contribute a further 1.27 ms.

The R50 residual differs by 15.54 ms, approximately 48% of the end-to-end difference. Most of this difference lies in the inter-stage remainder, at 22.2 ms for BEVDet-OCC and 7.0 ms for FlashOCC. The other reported component medians are close (Table 7). The breakdown identifies where the reported times differ.

At Swin-Base scale, S4+S6 differ by 17.11 ms, and accounting for the 1.09 ms increase in neck time gives a net S4–S6 difference of 16.02 ms. The total difference is 20.19 ms: the 16.60 ms residual difference is partly offset by -12.42 ms across S1–S3. The S4+S6 difference and the net BEV-side difference are each similar across scales, while the end-to-end differences are 32.11 and 20.19 ms.

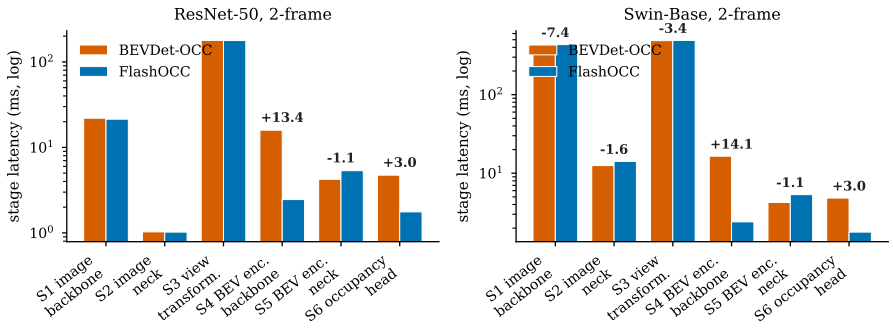

Figure 2: Stage medians for the two-frame configurations at R50 and Swin-Base scales. FlashOCC has lower S4 and S6 times and higher S5 time at both scales. S1–S3 also differ, particularly in the Swin-Base profiles. The bars compare configurations without isolating the effect of component substitution.

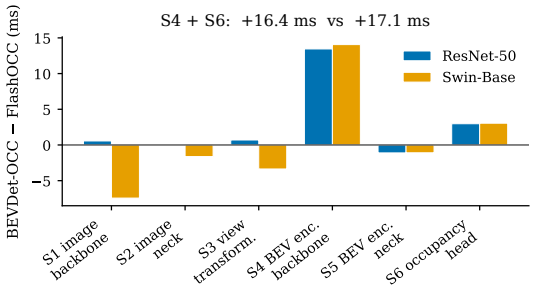

Figure 3: Per-stage differences (BEVDet-OCC minus FlashOCC) on a common linear axis. The S4+S6 annotation combines the BEV encoder backbone and occupancy head: +16.42 ms at R50 and +17.11 ms at Swin-Base. Including the negative S5 difference gives net BEV-side differences of 15.31 and 16.02 ms. Neither sum includes the front end or residual.

The stage breakdown therefore distinguishes similar local reductions from different total reductions. Expressed relative to BEVDet-OCC latency, the totals correspond to about 13% and 1.9%. These percentages depend on the front-end costs and residual as well as the BEV-side stages.

Relating stage costs to BEV-side components. Table 2 associates the three measured BEV-side stages with the component classes in the two model variants. The table identifies these components without enumerating every configuration difference.

Component	BEVDet-OCC	FlashOCC	Δ (R50, ms)
`img_bev_encoder_backbone`	`CustomResNet3D` (3D convs)	`CustomResNet` (2D)	+13.45 ms
`img_bev_encoder_neck`	`LSSFPN3D`	`FPN_LSS`	−1.11 ms
`occ_head`	`BEVOCCHead3D`	`BEVOCCHead2D`	+2.97 ms

Table 2: BEV-side component classes and the reported R50 stage differences, with Δ defined as BEVDet-OCC minus FlashOCC. Other configuration differences are not enumerated here.

The stage breakdown reveals an offset hidden by the total. The configurations with the 2D backbone and head have lower S4 and S6 times, but their `FPN_LSS` neck has higher S5 time than `LSSFPN3D`. Grouping all three stages would obscure this distinction.

4.4 SCALING BEHAVIOUR

Single-frame and two-frame stereo profiles. At R50 scale, view transformation contributes about 11% in the single-frame FlashOCC configurations, compared with 71.2% in two-frame BEVDet-OCC and 81.4% in two-frame FlashOCC. The image-backbone shares in the two-frame profiles are 8.8% and 9.8%, respectively. Repeated stage invocations require accumulation across calls, but do not fully explain this redistribution of cost. The two-frame configurations also introduce stereo estimation, so the measurements reflect both changes together.

Backbone and input resolution. The Swin-Base configurations use 512×1408 inputs, compared with 256×704 for R50. Across these settings, image-backbone time rises from about 22 ms to 426–433 ms and view-transformation time from about 178 ms to 487–490 ms. The two stages are co-dominant at Swin-Base scale. Their respective shares are 40.9% and 46.8% in BEVDet-OCC, and 42.5% and 48.0% in FlashOCC. Total latency is about 1040 and 1020 ms, respectively, with peak memory of 6.5 and 4.8 GB and residual shares of 8.6% and 7.2%. Head time remains 4.8 and 1.8 ms, or 0.5% and 0.2% of the respective totals.

4.5 SENSITIVITY AND REPEATABILITY

Repeated runs. Table 11 reports three additional runs per model at each scale. The S4+S6 differences range from 16.79 to 17.35 ms at R50 and from 17.75 to 17.83 ms at Swin-Base, giving ranges of 0.56 and 0.08 ms across the repeats. The end-to-end differences span approximately 2.71 and 14.39 ms, respectively, using the displayed medians. The local differences vary less across these repeats, although they also differ from the main-comparison values of 16.42 and 17.11 ms. These ranges characterise the observed repeatability.

Stage-boundary synchronisation. In Table 12, the largest positive difference in end-to-end median between synchronised and non-synchronised hooks is 1.37 ms; an earlier run recorded 1.87 ms. Both Swin-Base changes are negative. As detailed in §3, the ablation omits the main harness's enclosing function wrappers. The net differences quantify sensitivity in this setup.

Distinct inputs. Table 13 compares repeated-input timing with 50 distinct inputs in the existing ablation runs. For single-frame M1, end-to-end p95 changes from 39.88 to 53.01 ms; for M0, it changes from 41.12 to 46.56 ms, increases of 33.0% and 13.2%. The multi-frame changes are smaller and have mixed signs. The single-frame results are sensitive to the input-sampling protocol.

4.6 ACCURACY UNDER A BUDGET

Table 9 reports accuracy, end-to-end latency and peak memory together. The supplementary configuration notes record the evaluation-setting adjustment and checkpoint-key remapping used for BEVDet-OCC. Applying the median-latency rule of §3 gives the highest admissible accuracies of 32.01, 38.07 and 43.45 mIoU at budgets of 25, 255 and 1030 ms, respectively. Raising the budget to 1040 ms admits BEVDet-OCC Swin-Base without improving the best accuracy (Table 10).

The 43.45 mIoU configuration has about $41\times$ the median latency and $26\times$ the p95 of the 32.01 mIoU configuration in this measurement set. Median admission does not impose a tail-latency constraint: at the 25 ms budget, the selected model's p95 is 38.94 ms. Memory can also change the selection; imposing a 1500 MB cap at 255 ms leaves M0 as the best admissible configuration, at 31.95 mIoU.

A configuration near a budget threshold can change admission status across sessions. BEVDet-OCC Swin-Base meets the 1040 ms budget at 1039.82 ms in Table 9, but exceeds it in all three later repeats, at 1053.9–1055.7 ms (Table 11). This change illustrates why median-based admission is conditional on the measurement session and cannot guarantee a deployment deadline.

5 CONCLUSION

We presented a stage-level analysis of six BEVDet-OCC and FlashOCC configurations on Occ3D-nuScenes, combining six named measurement regions, accumulation over repeated module calls,

median and p95 latency, and explicit residual reporting. The resulting profiles show how the distribution of inference cost differs across configurations: the image backbone dominates single-frame FlashOCC, view transformation dominates the two-frame R50 stereo configurations, and both stages account for the largest costs at Swin-Base scale. The occupancy head contributes only 0.2–7.1% of total median latency in the measured configurations. These findings indicate that optimisation priorities depend on the full computation path.

Comparing the two model variants further reveals why local cost reductions and end-to-end speedups must be distinguished. FlashOCC's BEV encoder backbone and occupancy head together take 16.42 and 17.11 ms less at the two scales, while its slower neck reduces the net BEV-side differences to 15.31 and 16.02 ms. Front-end costs and the residual account for the remaining end-to-end difference. Across the additional runs, the combined backbone-and-head difference varies less than the total difference, although its values also shift relative to the main comparison. Thus, the measurements reveal similar local reductions at the two scales without establishing a scale-invariant architectural saving.

These patterns identify where to investigate optimisation, while the comparison and measurement boundaries determine how they can be interpreted. Temporal processing changes together with stereo estimation, backbone changes together with input resolution, and single-frame BEVDet-OCC remains unmeasured. The residual may include auxiliary learned paths and runtime work outside the instrumented modules, and its operational composition remains partly unresolved. The synchronisation ablation quantifies sensitivity under its own setup without validating every stage boundary or bounding all instrumentation effects. Missing configuration snapshots and incomplete links between tables and measurement runs also limit exact reconstruction.

Finally, an explicit median-latency budget connects these measurements to model selection by identifying the highest measured accuracy among admissible configurations. This selection remains conditional on the evaluated configuration set, runtime and measurement session. Together, stage decomposition and budget-explicit reporting provide a concrete basis for interpreting where inference time is spent and which measured accuracy–latency trade-off meets a stated constraint.

REFERENCES

Jiawei Hou, Xiaoyan Li, Wenhao Guan, Gang Zhang, Di Feng, Yuheng Du, et al. FastOcc: Accelerating 3D occupancy prediction by fusing the 2D bird's-eye view and 3D voxel. *arXiv preprint arXiv:2403.02710*, 2024. doi: 10.48550/arXiv.2403.02710.

Junjie Huang and Guan Huang. BEVDet4D: Exploit temporal cues in multi-camera 3D object detection. *arXiv preprint arXiv:2203.17054*, 2022. doi: 10.48550/arXiv.2203.17054.

Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. BEVDet: High-performance multi-camera 3D object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. doi: 10.48550/arXiv.2112.11790.

Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3D semantic occupancy prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. doi: 10.48550/arXiv.2302.07817.

Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. BEVStereo: Enhancing depth estimation in multi-view 3D object detection with dynamic temporal stereo. *arXiv preprint arXiv:2209.10248*, 2022. doi: 10.48550/arXiv.2209.10248.

Zhiqi Li, Zhiding Yu, David Austin, Mingsheng Fang, Shiyi Lan, Jan Kautz, et al. FB-OCC: 3D occupancy prediction based on forward-backward view transformation. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2023. doi: 10.48550/arXiv.2307.01492.

Haisong Liu, Yang Chen, Haiguang Wang, Zetong Yang, Tianyu Li, Jia Zeng, et al. Fully sparse 3D occupancy prediction. In *European Conference on Computer Vision (ECCV)*, 2024. doi: 10.48550/arXiv.2312.17118.

Bingqian Lu, Jianyi Yang, Weiwen Jiang, Yiyu Shi, and Shaolei Ren. One proxy device is enough for hardware-aware neural architecture search. *arXiv preprint arXiv:2111.01203*, 2021. doi: 10.48550/arXiv.2111.01203.

Jiaqi Peng, Tai Wang, Jiangmiao Pang, and Yuan Shen. Towards latency-aware 3D streaming perception for autonomous driving. *arXiv preprint arXiv:2504.19115*, 2025. doi: 10.48550/arXiv.2504.19115.

Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In *European Conference on Computer Vision (ECCV)*, 2020. doi: 10.48550/arXiv.2008.05711.

Armin Runge, Thomas Wenzel, Dimitrios Bariamis, Benedikt Sebastian Staffler, Lucas Rego Drumond, and Michael Pfeiffer. Bosch deep learning hardware benchmark. *arXiv preprint arXiv:2008.10293*, 2020. doi: 10.48550/arXiv.2008.10293.

Hao Shi, Ze Wang, Shangwei Guo, Mengfei Duan, Song Wang, Teng Chen, Kailun Yang, Lin Wang, and Kaiwei Wang. OneOcc: Semantic occupancy prediction for legged robots with a single panoramic camera. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2511.03571v2.

Chonghao Sima, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, et al. Scene as occupancy. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. doi: 10.48550/arXiv.2306.02851.

Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, et al. Occ3D: A large-scale 3D occupancy prediction benchmark for autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. doi: 10.48550/arXiv.2304.14365.

Yuping Wang, Xiangyu Huang, Xiaokang Sun, Mingxuan Yan, Shuo Xing, Zhengzhong Tu, et al. UniOcc: A unified benchmark for occupancy forecasting and prediction in autonomous driving. *arXiv preprint arXiv:2503.24381*, 2025. doi: 10.48550/arXiv.2503.24381.

Yuqi Wang, Yuntao Chen, Xingyu Liao, Lue Fan, and Zhaoxiang Zhang. PanoOcc: Unified occupancy representation for camera-based 3D panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. doi: 10.48550/arXiv.2306.10013.

Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. SurroundOcc: Multi-camera 3D occupancy prediction for autonomous driving. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. doi: 10.48550/arXiv.2303.09551.

Haoran Xu, Peixi Peng, Xinyi Zhang, Guang Tan, Yaokun Li, Shuaixian Wang, and Luntong Li. Exploiting continuous motion clues for vision-based occupancy prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8860–8868, 2025. doi: 10.1609/aaai.v39i8.32958.

Zichen Yu, Changyong Shu, Jiajun Deng, Kangjie Lu, Zongdai Liu, Jiangyong Yu, et al. FlashOcc: Fast and memory-efficient occupancy prediction via channel-to-height. *arXiv preprint arXiv:2311.12058*, 2023. doi: 10.48550/arXiv.2311.12058.

Yanan Zhang, Jinqing Zhang, Zengran Wang, Junhao Xu, and Di Huang. Vision-based 3D occupancy prediction in autonomous driving: a review and outlook. *arXiv preprint arXiv:2405.02595*, 2024. doi: 10.48550/arXiv.2405.02595.

A FULL PER-STAGE NUMBERS

This appendix collects the reported stage statistics, invocation counts, residual breakdown, accuracy, latency and memory summaries, together with the sensitivity and repeat measurements. Configuration abbreviations: **FO** = FlashOCC, **BD** = BEVDet-OCC; **1f/2f** = single-frame / two-frame stereo; **R50/SwB** = ResNet-50 / Swin-Base backbone.

A.1 PER-STAGE MEDIANS

Stage (median, ms)	FO-1f-M1	FO-1f-M0	BD-2f-R50	FO-2f-R50	BD-2f-SwB	FO-2f-SwB
S1 image backbone	10.69	10.70	21.95	21.39	425.55	432.99
S2 image neck	0.51	0.51	1.03	1.02	12.50	14.12
S3 view transformation	2.88	2.28	178.40	177.71	486.56	489.92
S4 BEV encoder backbone	2.41	2.68	15.90	2.46	16.46	2.40
S5 BEV encoder neck	4.98	1.50	4.23	5.34	4.25	5.34
S6 occupancy head	1.77	0.71	4.74	1.76	4.83	1.78
sum of stages	23.25	18.38	226.25	209.67	950.15	946.56
residual	1.61	1.60	24.18	8.64	89.67	73.07
end-to-end	**24.87**	**19.98**	**250.43**	**218.31**	**1039.82**	**1019.63**
peak memory (MB)	1625	1336	3736	1965	6514	4806

Table 3: Stage and end-to-end medians across all six configurations. The stage sum is the sum of the six stage medians, and the residual is the end-to-end median minus that sum, subject to rounding. Memory is peak allocated GPU memory in decimal MB. Initial input transfer is excluded from timing.

A.2 PER-STAGE P95

Stage (p95, ms)	FO-1f-M1	FO-1f-M0	BD-2f-R50	FO-2f-R50	BD-2f-SwB	FO-2f-SwB
S1 image backbone	11.11	23.67	22.25	21.44	429.03	434.22
S2 image neck	0.53	0.53	1.05	1.04	13.31	14.39
S3 view transformation	6.79	2.31	179.51	178.74	488.81	491.55
S4 BEV encoder backbone	2.48	2.70	16.34	2.50	16.92	2.48
S5 BEV encoder neck	8.59	1.51	4.26	5.37	4.29	5.40
S6 occupancy head	1.79	0.73	5.15	1.77	5.48	1.80
end-to-end	38.94	42.49	251.55	220.13	1055.83	1026.79

Table 4: Per-stage and end-to-end p95 across all six configurations when repeatedly timing a fixed input. M0 has a large S1 p95-to-median difference (23.67 versus 10.70 ms); M1 has larger differences in S3 and S5. These statistics describe repeated execution of the same input, not variation across scenes. Stage p95 values need not sum to end-to-end p95.

A.3 CONFIGURATION COMPARISON AT SWIN-BASE SCALE

Stage	BEVDet-OCC (median, ms)	FlashOCC (median, ms)	Δ
S1 image backbone	425.55	432.99	−7.44
S2 image neck	12.50	14.12	−1.62
S3 view transformation	486.56	489.92	−3.36
S4 BEV encoder backbone	**16.46**	**2.40**	**+14.06**
S5 BEV encoder neck	4.25	5.34	−1.09
S6 occupancy head	**4.83**	**1.78**	**+3.05**
sum of stages	*950.15*	*946.56*	*+3.59*
residual	*89.67*	*73.07*	*+16.60*
end-to-end	**1039.82**	**1019.63**	**+20.19**
peak memory (MB)	6514	4806	+1708

Table 5: Two-frame Swin-Base configuration comparison at 512×1408 input. Differences use BEVDet-OCC minus FlashOCC, including memory. The stage differences are plotted in Figure 3; their sum and the residual difference together account for the end-to-end difference up to rounding.

A.4 INVOCATIONS PER FORWARD PASS

Invocations per forward	FO-1f-M1	FO-1f-M0	BD-2f-R50	FO-2f-R50	BD-2f-SwB	FO-2f-SwB
S1 image backbone	1	1	2	2	2	2
S2 image neck	1	1	2	2	2	2
S3 view transformation	1	1	2	2	2	2
S4 BEV encoder backbone	1	1	1	1	1	1
S5 BEV encoder neck	1	1	1	1	1	1
S6 occupancy head	1	1	1	1	1	1

Table 6: Recorded module invocations per forward pass. S1–S3 are called twice in the two-frame configurations, while S4–S6 are called once. Stage times accumulate both calls; retaining only the last would omit earlier work. The counts cover instrumented module calls, leaving coverage of auxiliary or partial-module execution paths unverified.

A.5 RESIDUAL BREAKDOWN

Residual summary (ms)	FO-1f-M1	FO-1f-M0	BD-2f-R50	FO-2f-R50	BD-2f-SwB	FO-2f-SwB
input preparation	0.24	0.24	0.50	0.51	0.53	0.53
post-processing	1.08	1.05	1.06	1.04	1.09	1.25
framework	0.04	0.04	0.06	0.06	0.07	0.06
inter-stage remainder	0.25	0.27	22.20	7.03	87.91	71.08
total residual	1.61	1.60	24.18	8.64	89.67	73.07

Table 7: Residual breakdown in milliseconds. Input preparation is the time spent in `prepare_inputs`. The post-processing bucket subtracts S6 from the time in `simple_test_occ` and does not isolate argmax. The framework bucket is end-to-end time minus `simple_test`. The inter-stage remainder is the time in `simple_test` left after subtracting input preparation, S1–S6 and post-processing. The harness clips these three derived buckets at zero per iteration before taking their medians. Total residual follows the main harness's difference-of-medians convention: end-to-end median minus the sum of the six stage medians (§3). It is not the median of per-iteration residuals. Because of aggregation, clipping and rounding, the component medians need not sum to the total (largest displayed discrepancy 0.36 ms). Initial input transfer is outside the timed path.

A.6 SINGLE-FRAME STAGE DECOMPOSITION

Stage	M1 (median, ms)	share	M0 (median, ms)	share
S1 image backbone	10.69	43.0%	10.70	53.6%
S2 image neck	0.51	2.1%	0.51	2.6%
S3 view transformation	2.88	11.6%	2.28	11.4%
S4 BEV encoder backbone	2.41	9.7%	2.68	13.4%
S5 BEV encoder neck	4.98	20.0%	1.50	7.5%
S6 occupancy head	1.77	7.1%	0.71	3.6%
sum of stages	*23.25*	*93.5%*	*18.38*	*92.0%*
residual	*1.61*	*6.5%*	*1.60*	*8.0%*
end-to-end	**24.87**		**19.98**	
peak memory (MB)	1625		1336	

Table 8: Per-stage decomposition of the single-frame configurations. Values are medians over 50 iterations at batch size 1 on the L40 in FP32.

A.7 ACCURACY, LATENCY AND MEMORY

Config.	mIoU	mIoU (pub.)	median (ms)	p95 (ms)	mem (MB)	budget (ms)
FlashOCC R50 1f (M1)	32.01	32.08	24.87	38.94	1625	25
FlashOCC R50 1f (M0)	31.95	31.95	19.98	42.49	1336	25
FlashOCC R50 2f	38.07	37.84	218.31	220.13	1965	255
BEVDet-OCC R50 2f	36.01	36.01	250.43	251.55	3736	255
BEVDet-OCC Swin-B 2f	42.02	42.45	1039.82	1055.83	6514	1040
FlashOCC Swin-B 2f	43.45	43.52	1019.63	1026.79	4806	1030

Table 9: Accuracy and measured cost. Accuracy uses the validation split, while median and p95 latency use repeated fixed-input timing (§3). The budget column gives the illustrative median-latency tier used in Table 10; memory is in decimal MB. Published reference values are recorded in the released configuration files, except for FlashOCC Swin-B: its 43.52 reference comes from a sibling configuration with different learning-rate and loss settings. Absolute accuracy differences are 0.07, 0.00, 0.23, 0.00, 0.43 and 0.07 mIoU points. The R50 rows use 256×704 inputs and the Swin-Base rows 512×1408. Our evaluations use FP32; the recorded reference values do not specify precision.

A.8 ADMISSIBLE CONFIGURATIONS UNDER LATENCY BUDGETS

Budget (median, ms)	Admissible configurations	Best mIoU	median / p95 of the best (ms)
25	FO-1f-M1, FO-1f-M0	32.01	24.87 / 38.94
255	+ FO-2f-R50, BD-2f-R50	38.07	218.31 / 220.13
1030	+ FO-2f-SwB	43.45	1019.63 / 1026.79
1040	+ BD-2f-SwB (best unchanged)	43.45	1019.63 / 1026.79

Table 10: Cumulative admissible sets obtained from the medians in Table 9; + denotes newly admitted configurations. Budgets are illustrative thresholds, and p95 is reported alongside rather than used for admission. At 25 ms the selected model's p95 exceeds the budget. At 1040 ms BD-2f-SwB meets the median threshold in this measurement set but not in the later repeats (Table 11). With an additional 1500 MB memory cap, M0 is the best admissible configuration at either 25 or 255 ms, at 31.95 mIoU.

A.9 RUN-TO-RUN REPEATABILITY

Scale	Repeat	FO median (ms)	FO p95 (ms)	BD median (ms)	BD p95 (ms)	S4+S6 (ms)
R50	1	219.79	221.53	252.08	254.06	16.79
R50	2	218.60	219.94	252.50	253.98	17.02
R50	3	218.52	219.84	253.52	254.94	17.35
Swin-B	1	1004.02	1011.12	1053.91	1061.28	17.83
Swin-B	2	1019.06	1028.69	1055.71	1064.33	17.81
Swin-B	3	1019.90	1028.36	1055.40	1060.65	17.75

Table 11: Three additional runs per model and scale, in milliseconds. The launch scripts alternate order across R50 pairs (FO, BD; BD, FO; FO, BD) and use FO, BD for each Swin-B pair. This ordering samples repeated execution but does not rule out device-state or order effects. The S4+S6 column is the BEVDet-OCC minus FlashOCC difference, with spans of 0.56 ms at R50 and 0.08 ms at Swin-B. FlashOCC's end-to-end median spans 1.27 and 15.88 ms, respectively. In this later session, BD-2f-SwB takes 1053.9–1055.7 ms, compared with 1039.82 ms in Table 9. Ranges describe these runs, not confidence intervals or a bound on future variation.

A.10 SYNCHRONISATION ABLATION

Config.	A: sync. (median, ms)	B: no sync. (median, ms)	A−B (ms)	A−B (%)
FO-1f-M1	25.54	24.79	0.74	2.91
FO-1f-M0	19.50	18.80	0.70	3.61
FO-2f-R50	218.12	216.75	1.37	0.63
BD-2f-R50	252.67	252.11	0.56	0.22
FO-2f-SwB	1012.90	1017.35	−4.44	−0.44
BD-2f-SwB	1054.54	1054.67	−0.13	−0.01

Table 12: Synchronisation ablation. Mode A synchronises at each stage-hook boundary; mode B retains the hooks but omits their synchronisation. Both modes synchronise around the end-to-end timer and omit the three enclosing function wrappers used by the main protocol (§3). Thus A is not an exact replication of that protocol. The largest positive end-to-end median difference is 1.37 ms in this run; an earlier run measured 1.87 ms for FO-2f-R50. The negative differences for both Swin-B configurations show that the net difference also reflects execution variability. These observations quantify sensitivity in the tested setup. They neither provide a strict upper bound on instrumentation overhead nor establish the operations underlying the residual.

A.11 P95 ACROSS DISTINCT FRAMES

Configuration	p95, one frame × 50	p95, 50 distinct frames	change
FO-1f-M1	39.88	53.01	+33.0%
FO-1f-M0	41.12	46.56	+13.2%
FO-2f-R50	219.19	224.48	+2.4%
BD-2f-R50	253.96	252.39	−0.6%
FO-2f-SwB	1020.09	1018.94	−0.1%
BD-2f-SwB	1058.56	1049.30	−0.9%

Table 13: End-to-end p95 in milliseconds under repeated-input and distinct-input timing, using the same ablation harness. The repeated-input mode reuses the first loader sample; the distinct-input mode consumes 10 samples during warm-up and times the next 50 with loading and transfer excluded. These runs are separate from Table 9; their single-frame repeated-input p95 values are 39.88/41.12 ms, compared with 38.94/42.49 ms there. Single-frame p95 is higher with distinct inputs in these runs, while multi-frame changes have mixed signs. The summaries do not identify which stage contributes to the difference.

A.12 PUBLISHED EFFICIENCY FIGURES AND MEASUREMENT SETTINGS

Method	Reported figure	Hardware	Precis.	Runtime	Granularity
FlashOCC (Yu et al., 2023)	$7.5 \rightarrow 3.1$ ms, $398 \rightarrow 124$ MiB	RTX 3090	FP16	TensorRT	BEV enc. + head
FastOcc (Hou et al., 2024)	221 ms at 40.75 mIoU	four V100	not stated	PyTorch	2D / 2D-to-3D / 3D
FullySparse (Liu et al., 2024)	25.4 FPS (lite, 1f)	V100	not stated	PyTorch	end-to-end
PanoOcc (Wang et al., 2024)	203 ms (Base, 71.7 mIoU)	A100	not stated	—	end-to-end

Table 14: Selected published efficiency figures and their reported settings. Precision marked "not stated" refers to the cited runtime figure, not necessarily to every experiment in that paper. These figures provide context for the literature comparison. Differences in their measurement protocols preclude a matched ranking with our results. Embedded occupancy studies are discussed separately in §2.

B AI USE STATEMENT

In this study, we used generative AI tools to search and summarise the literature, write a small amount of experimental code, and polish the language of the manuscript. All substantive content and scientific claims were created entirely by the authors. We have reviewed all AI-assisted work to ensure accuracy and originality, and take full responsibility for the content of the paper. LLMs are not listed as authors.