PAVR: On-Device Food Nutrition Estimation via Plate-Anchored Volumetric Reasoning

Haruto Fujita[1] and Keiji Yanai[1]

The University of Electro-Communications, Tokyo, Japan
fujita-h@mm.inf.uec.ac.jp, yanai@cs.uec.ac.jp

Abstract. Estimating the energy and macronutrient content of a meal from a single photograph requires not only food recognition, but also an absolute estimate of portion size. Relative monocular depth can supply geometric cues to a multimodal large language model (MLLM), yet it typically needs an extra depth network, a per-image scale assumption, and memory that is difficult to keep coresident with a mobile vision-language model. We propose *Plate-Anchored Volumetric Reasoning* (PAVR), a lightweight geometry front-end that converts a metric depth map and plate/food masks into four text descriptors: plate diameter, food area, and the maximum and mean food heights above a fitted plate plane. An MLLM then receives only the RGB image and this short prompt; depth maps and masks are never passed to the language model, and no explicit volume formula or food-category regressor is provided. On a fixed Nutrition5k test split of 346 dishes, LoRA-tuned models with PAVR match RGB-only fine-tuning on 4B-scale backbones and improve calorie mean absolute error (MAE) by 19.9% for Qwen2-VL-2B. Relative to a monocular-depth text baseline, PAVR attains comparable calorie MAE without a second vision network. We also list published Nutrition5k results with backbone sizes as reference points, since evaluation protocols differ. We further deploy the pipeline on an iPhone with LiDAR, showing that metric plate-anchored geometry can be computed on device and consumed by a quantized MLLM.

Keywords: Food nutrition estimation · RGB-D · LiDAR · multimodal large language models · on-device inference

1 Introduction

Automated dietary assessment from food photographs is a long-standing computer vision problem [10, 22]. A practical system must identify what is on the plate *and* how much of it is there. Appearance cues alone are often insufficient: the same recipe photographed from a different distance, or served in a different portion, can yield similar RGB statistics while differing substantially in energy. Prior mobile systems therefore introduced an explicit size cue, either through AR markers [21], geometric food models [3], depth cameras [2], or smartphone LiDAR with category-wise volume-to-mass regressors [4].

Multimodal large language models (MLLMs) change this design space. Instead of training a closed-set calorie regressor, one can prompt a vision-language model to emit structured nutrition estimates for unseen dishes [19, 25]. CalorieVoL [19] showed that injecting a volumetric summary derived from monocular depth into the prompt improves calorie estimation over RGB-only MLLMs. That approach, however, still depends on a dedicated depth estimator, and the estimated depth is relative: converting it into a physical quantity requires a scale assumption such as a known camera-to-plate distance.

On a smartphone these extra ingredients are costly. A 4B-class vision-language model already occupies most of the resident memory budget. Published MLLM estimators on Nutrition5k are larger still: the fine-tuned models use 7B–13B backbones [20, 25]. Keeping a monocular depth network coresident, or reloading it on every capture, is unattractive for an interactive food log. Meanwhile, recent iPhones expose metric depth through their LiDAR sensors. The sensor is not a substitute for recognition, but it does provide an absolute millimetre scale without a learned depth model.

This paper asks whether that metric geometry can be compressed into a few text tokens that an MLLM can use, without forcing the model to ingest a depth map or to apply a category-specific volume formula. We call the resulting front-end *Plate-Anchored Volumetric Reasoning* (PAVR). A plate is treated as a locally planar spatial anchor. From RGB-D we extract four quantities: plate diameter, food area, and the 95th-percentile and mean heights of food above the plate plane. These four numbers are written into a short prompt. The MLLM, fine-tuned with LoRA [7], outputs calories, mass, and macronutrients as JSON.

The design has three intended consequences. First, unknown dishes need not be mapped to a closed set of volume-to-mass regressors, unlike LiDARCalorieCam [4]. Second, the MLLM never sees the depth map, so the expensive language-model context remains an RGB image plus a few tokens. Third, on a LiDAR-equipped phone the same four numbers can be computed from sensor depth, avoiding a second neural network at inference time.

Our main empirical finding is more nuanced than a blanket accuracy claim. On Nutrition5k [22], with matched LoRA training from public base checkpoints, adding PAVR text substantially helps a 2B model and is essentially tied with RGB-only LoRA on 4B models. A separately trained monocular-depth text condition is also close in calorie MAE. We therefore do *not* claim that LiDAR is universally more accurate than monocular depth on this overhead RGB-D benchmark. We claim that plate-anchored metric descriptors are a compact, deployable alternative that preserves comparable nutrition MAE while removing the extra depth network and per-image scale assumption.

Our contributions are:

- PAVR, a plate-plane geometry front-end that turns metric depth and instance masks into four text descriptors for an MLLM.
- A controlled Nutrition5k study across Qwen2-VL-2B, Qwen3-VL-4B, and Gemma 4 E2B/E4B, with a fixed 346-dish test split and fully parsed JSON outputs.

– Diagnostic evidence, from a probe on the monocular-depth-text adapter (Cond3), that a LoRA-tuned MLLM does use numeric prompt text (shuffling it changes most predictions) even when aggregate MAE is stable, together with preliminary on-device measurements of the iPhone pipeline.
– A reference comparison with published Nutrition5k systems that lists backbone size next to calorie MAE and flags where evaluation sets differ.

2 Related Work

2.1 Image-Based Nutrition Estimation and Nutrition5k

Im2Calories [10] combined food segmentation, 3D volume, and a nutrition database for a mobile food diary. Subsequent work explored portion cues from geometric primitives [3], AR scale [21], smartphone photographs with reference objects [1,11], and dedicated food segmentation datasets [9,12]. Nutrition5k [22] then released a large RGB-D corpus of cafeteria dishes and became the common benchmark for absolute nutrition regression. Its baselines show where the difficulty lies: an InceptionV2 regressor has a 9.5% relative error for calories *per gram* but 26.1% (70.6 kcal MAE) for absolute calories from RGB, and overhead depth lowers the MAE to 47.6 kcal (41.3 kcal with a volume scalar) [22]. Portion estimation is thus the dominant difficulty, and metric depth the most effective cue tested there. Later work mostly refines RGB-D fusion: Swin-Nutrition [14], a multi-scale RGB-D fusion network [15], DPF-Nutrition [5] (predicted depth fused with RGB; 37.9 kcal MAE, 14.7% PMAE), and single-view shape reconstruction [16]. NutritionVerse [17] compared several dietary-intake estimators and emphasized that portion size remains a dominant error source. These are trained regression networks, several report percentage errors (PMAE) rather than kcal, and the RGB-D variants rely on the fixed overhead RealSense rig. Because test sets differ in mean calories, we report calorie PMAE next to kcal MAE for our own results and, where the cited paper gives it, for published ones (Sec. 4.9).

2.2 Depth and Volume-Based Mobile Systems

Depth cameras make the missing scale observable. DepthCalorieCam [2] estimated food volume from a mobile depth camera and converted volume to calories with category-dependent models; its successful experiments cover only three food categories [22]. Fujita and Yanai [4] used iPhone LiDAR point clouds, fused several geometric volume estimators, and applied per-category linear maps $W = aV + b$. That pipeline attains low error on a small set of known foods, but it cannot be applied unchanged to an open set of cafeteria dishes, and the original system offloaded part of the computation. Our work keeps the LiDAR scale cue but discards the category-wise formula: the MLLM must infer nutrition from appearance plus four geometric numbers.

CalorieVoL [19] is the closest MLLM baseline. It segments food, estimates monocular depth, and injects an integrated volume into the prompt. The original

implementation uses a learned depth model and a dish-relative height obtained from depth differences. We treat that idea as a comparison condition (Cond3 in Sec. 4.2), re-implemented with YOLO masks and Depth Anything V2 [24], rather than as a bit-exact reproduction of CalorieVoL. CalorieVoL and its journal extension [19,20] evaluate the idea on zero-shot models, and the effect of the injected volume is mixed: it lowers GPT-4o's calorie MAE from 82.7 to 78.8 kcal, raises GPT-4V's from 80.7 to 83.8 kcal, and, for LLaVA-1.5-13B, raises it from 109.6 to 6122.7 kcal ($r = -0.04$), i.e., the model fails with the injected number [20]. A volume number in the prompt thus helps only when the model can use it, which motivates training on the numeric text, as we do with LoRA.

2.3 Multimodal LLMs for Food

Instruction-tuned vision-language models [8, 13, 23] can follow open-ended prompts and emit structured text. FoodLMM [25], built on the 7B-scale LISA model, specializes this paradigm to food assistance and notes that quantity estimation still needs a scale signal. We follow that observation: PAVR is not a new MLLM architecture, but a geometry-to-text interface that existing backbones can consume after lightweight LoRA adaptation [7]. CalorieLLaVA [18, 20] instead LoRA-tunes LLaVA-1.5 on Nutrition5k image–calorie pairs (64.3 kcal MAE at 13B and 74.2 kcal at 7B, on the authors' 506-image test split). These are the closest published MLLM estimators to ours; their 7B–13B or proprietary backbones are beyond what an on-device setting can host (Sec. 4.9).

3 Method

3.1 Overview

Figure 1 summarizes the pipeline. The capture stage provides an RGB image and a metric depth map: Intel RealSense D435 overhead frames when training and evaluating on Nutrition5k, and iPhone LiDAR when running on a device. An instance segmentation model predicts plate and food masks from RGB. PAVR then combines the masks with depth to produce four scalar descriptors. An MLLM is prompted with the RGB image and a short English string that embeds those descriptors, and it returns a JSON object with energy, mass, protein, fat, and carbohydrate.

Two implementation facts are essential for interpreting the experiments. The MLLM input is *only* RGB plus text; depth and masks are not visual tokens. Nutrition5k depth is structured-light stereo, not LiDAR, so we refer to it as RealSense metric depth and reserve the word LiDAR for the iPhone sensor.

3.2 Instance Segmentation

We fine-tune an Ultralytics YOLO26-seg model [**?**] for two classes (plate, food) on UEC-FoodPIX Complete [12] extended with plate annotations, and apply it

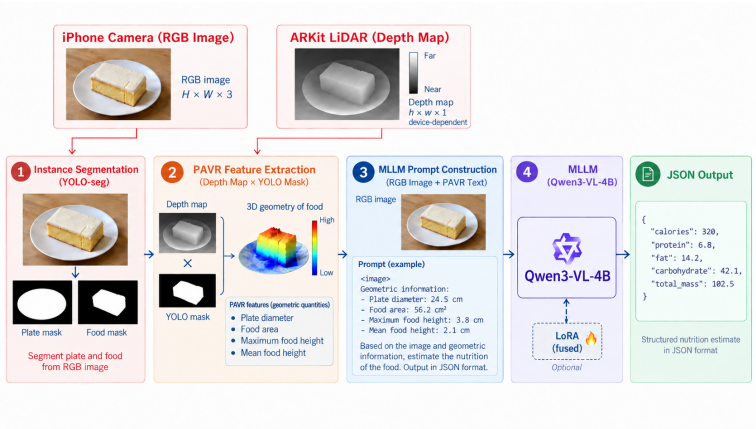


Fig. 1: PAVR pipeline. RGB and a metric depth map are captured together. YOLO predicts plate and food masks; PAVR converts depth and masks into four geometric quantities; an MLLM consumes the RGB image and a text prompt and emits structured nutrition estimates. LoRA is optional at training time and fused for on-device deployment.

to the Nutrition5k overhead frames. Depth-based ellipse fitting of the plate is unreliable on Nutrition5k's square trays and irregular dishware, so we rely on learned masks. If the plate mask is a thin rim around the food, we fill interior holes before geometry extraction so that food pixels inside the rim are treated as lying on the plate support. The inner plate region used for plane fitting is a morphological erosion of this filled mask (6 px, elliptical kernel), which reduces leakage of food pixels into the plane estimate.

Food is represented as a single blob per tray. Multiple items on one cafeteria plate are therefore aggregated, which matches Nutrition5k's tray-level nutrition labels. The method does not currently emit per-item nutrition.

3.3 Plate-Anchored Geometry

Pinhole unprojection. Let $d(u, v)$ be metric depth in millimetres at pixel (u, v), and let (f_x, f_y) and (c_x, c_y) be the pinhole intrinsics (Fig. 2). A pixel back-projects to

$$X = \frac{(u - c_x)\, d}{f_x}, \qquad Y = \frac{(v - c_y)\, d}{f_y}, \qquad Z = d. \tag{1}$$

On Nutrition5k we use the RealSense D435 nominal calibration at 640×480, $f_x = f_y = 615$ and $(c_x, c_y) = (320, 240)$. On device we use the per-frame camera intrinsics.

Plate diameter and area. Plate contour pixels are unprojected with Eq. (1). The plate diameter is the maximum 3D Euclidean chord among a subsample of those

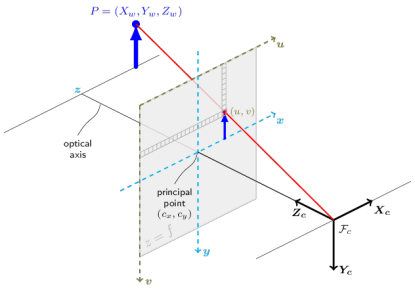


Fig. 2: Pinhole camera model used to convert a depth pixel (u, v) and metric depth Z into a 3D point $P = (X_w, Y_w, Z_w)$. Plate diameter is computed from pairwise 3D distances among plate-contour points.

points,

$$D_{\text{plate}} = \max_{i \neq j} \|\mathbf{p}_i - \mathbf{p}_j\|_2. \tag{2}$$

Each plate pixel with valid depth contributes an approximate ground-plane patch of area $d(u, v)^2/(f_x f_y)$. The same conversion, restricted to the food support, yields the food area A_{food} reported in cm^2.

Plate plane and food height. A few degrees of camera tilt induce a depth gradient across the plate (about 11 mm over an 180 mm plate at 3.4°). A scalar depth threshold would then confuse tilt with food height. We therefore fit a plane $z = au + bv + c$ by least squares to inner-plate pixels whose depth lies near the mode of the plate-depth histogram (inlier band 2–5 mm). Food height at a pixel is the signed difference between the plane and the measured depth,

$$h(u, v) = \big(au + bv + c\big) - d(u, v). \tag{3}$$

Pixels with $h(u, v) > 3$ mm form the depth-based food support. In the canonical protocol this support is intersected with the YOLO food mask. The two height descriptors are the 95th percentile of $\{h(u, v)\}$ (robust maximum) and the mean over the same set. Using a percentile rather than the literal maximum suppresses isolated depth spikes.

Why four numbers, not a single volume. A natural volume proxy is $V \approx A_{\text{food}}\bar{h}$. We deliberately do not compute V or supply Atwater factors. Mean height is the quantity that would enter that product; the 95th-percentile height preserves peak-shape information that averaging can dilute at mask boundaries; plate diameter is a scale anchor that is stable even when the food mask is imperfect. Early prototypes also injected camera distance and tilt. Those extra numbers behaved as noise in development runs, so the prompt keeps only

$$\mathbf{g} = \big(D_{\text{plate}},\ A_{\text{food}},\ h_{95},\ \bar{h}\big). \tag{4}$$

Equation (3) uses optical-axis depth difference, not distance along the plate normal. PAVR is therefore a tilt-*mitigating* representation, not a proof of viewpoint invariance. We do not claim that oblique handheld captures are already solved.

3.4 Prompting and Output

The user prompt is a single English line, for example:

```
[PAVR] plate_diameter:245mm food_area:56cm2
food_max_height:38mm food_mean_height:21mm
Estimate the nutritional content of this meal.
```

The system prompt asks for JSON of the form `{"kcal":..., "mass_g":..., "protein_g":..., "fat_g":..., "carb_g":...}`. We do not ask the model to name the food in the default protocol. Placing a predicted food name before the numbers can propagate a recognition error into the nutrient head; omitting the name performed best in a three-way format study on Qwen2-VL-2B (no name / name last / name first). All reported experiments use English prompts, which aligned better with the pretrained tokenizers than an otherwise identical Japanese template.

3.5 Fine-Tuning

Unless noted, we attach LoRA [7] of rank $r = 64$, $\alpha = 128$, and dropout 0.05 to the language-model projections, and train for two epochs with AdamW at 10^{-4} in bfloat16. Qwen2-VL-2B uses batch size 4; 4B-class models use batch size 2. The default seed for the matched comparison is 42. Gemma 4 wraps some linear layers in a custom class, so LoRA is restricted to language-model projections matched by name. Adapters are trained independently for each input condition from the public base checkpoint; we do not initialize one condition from another unless a subsection explicitly says so.

4 Experiments

4.1 Dataset and Split

We use Nutrition5k [22] overhead RGB-D frames from the RealSense D435. Starting from 3,490 labelled dishes we discard 1,457 frames: 1,020 in which no plate is detected (bowls, cups, and other non-plate ware) and 437 whose geometry estimate is implausible (unreadable frames, implausible plate diameter, no food above the plate, or outlier depth values that give an impossible table distance). Both decisions are automatic (the fine-tuned YOLO26-seg model of Sec. 3.2 and simple geometric checks), so they are not error-free: some frames may be wrongly kept or discarded. The remaining 2,033 dishes are split at random (seed 42) into 1,423 / 264 / 346 train / validation / test dishes. This split is not scan-disjoint: dishes from the same incremental scan of a plate [22] can fall on both sides, so

absolute errors may be optimistic relative to the official split. The test set is frozen for all primary tables.

This cohort is already biased toward plates that PAVR can describe: bowls are largely excluded, and Nutrition5k itself is captured from a fixed overhead rig [22]. Results should not be read as an in-the-wild handheld benchmark.

4.2 Conditions

We compare four input conditions; in every condition the MLLM sees RGB and text only, and depth maps never enter the model.

- **Cond1 (zero-shot RGB):** public instruction-tuned MLLM, RGB only, no LoRA.
- **Cond2 (RGB + LoRA):** same RGB input, LoRA-tuned on the nutrition JSON.
- **Cond3 (monocular depth text):** RGB + LoRA, with a numeric string derived from Depth Anything V2 [24] (relative depth, optionally globally calibrated) rather than RealSense.
- **Cond4 (PAVR):** RGB + LoRA, with the four RealSense (or, on device, LiDAR) descriptors of Eq. (4).

In every condition the MLLM sees RGB and text only. Cond3 is a *reference implementation* inspired by CalorieVoL [19], not a replica of that paper's GroundingDINO/SAM/Marigold stack. Relative-depth variants of Cond3 still require a scale story (a 350 mm centre-depth assumption in an older pipeline, or a train-only affine calibration in a later pipeline). PAVR does not. Likewise, Cond2 follows the recipe of CalorieLLaVA [18] (LoRA fine-tuning of a general MLLM to emit nutrition values), but on 2B–4B backbones and with JSON output. Cond2 and Cond3 are thus in-protocol counterparts of the two closest MLLM baselines, and they are the only comparisons in this paper that share a test set with PAVR.

4.3 Metrics

The primary metric is calorie MAE in kcal on JSON-parseable test outputs, together with the coefficient of determination R^2. Following Nutrition5k [22], we also report the percentage MAE, $\text{PMAE} = 100 \cdot \text{MAE}/\bar{y}$, where $\bar{y}$ is the mean ground-truth value of the same field in the test set (260.6 kcal for calories on our 346 test dishes); it allows comparison across test sets with different mean calories. Mass MAE (g) is reported for the matched comparison. Macronutrients are reported only for the multi-seed experiment of Sec. 4.7. Unless noted, every cell below parses 346/346 test dishes.

4.4 Matched RGB versus PAVR

Table 1 is the primary comparison. Each backbone is trained twice from the same public checkpoint, once on RGB-only JSON (Cond2) and once on RGB+PAVR

Table 1: Matched pure-base LoRA comparison on the frozen Nutrition5k test set ($n = 346$, parse 346/346). Cells give calorie MAE in kcal with calorie PMAE (%) in parentheses. Δ is Cond4−Cond2 calorie MAE; negative values favour PAVR. One seed (42).

	Cond2 RGB+LoRA		Cond4 PAVR			Mass
Model	MAE (PMAE)	R^2	MAE (PMAE)	R^2	Δ kcal	2→4 (g)
Qwen2-VL-2B	80.87 (31.0)	0.619	**64.82** (24.9)	0.779	−16.05 (−19.9%)	56.1→37.0
Qwen3-VL-4B	**62.49** (24.0)	0.782	62.73 (24.1)	0.789	+0.23 (+0.4%)	37.8→37.2
Gemma 4 E2B	**65.39** (25.1)	0.758	68.75 (26.4)	0.737	+3.36 (+5.1%)	41.1→41.8
Gemma 4 E4B	63.28 (24.3)	0.779	**61.02** (23.4)	0.794	−2.26 (−3.6%)	38.2→37.5

Table 2: Earlier-generation Cond3/Cond4 calorie MAE / R^2 on the same 346-dish test set. The Qwen3-VL-4B Cond4 value is not the matched pure-base result in Tab. 1.

Model	Cond3 (mono. depth text)	Cond4 (PAVR)
Qwen2-VL-2B	78.6 / 0.65	77.3 / 0.67
Qwen3-VL-4B	59.7 / 0.80	60.1 / 0.80
Gemma 4 E2B	76.6 / 0.68	68.0 / 0.75
Gemma 4 E4B	64.6 / 0.75	62.7 / 0.78

JSON (Cond4), with identical hyperparameters and the English food-name-free split.

PAVR is not uniformly better. The 2B Qwen2 model improves by 16.05 kcal (19.9%) and mass MAE drops by 19.1 g. On Qwen3-VL-4B the two conditions are tied to a fraction of a kilocalorie. Gemma 4 E4B improves modestly; Gemma 4 E2B is slightly worse with PAVR. We therefore treat PAVR as a high-value cue for capacity-constrained models, and as a near-neutral add-on for 4B RGB specialists, rather than as a guarantee of state-of-the-art MAE.

Zero-shot RGB (Cond1) is far worse on the same test IDs (calorie MAE 108.7 / 102.7 / 128.9 / 112.9 for the four backbones, with Gemma 4 E2B parsing 337/346). LoRA, not PAVR, is the dominant jump from a generic MLLM to a nutrition estimator.

4.5 Monocular Depth Text

A previous adapter generation, evaluated on the same 346 dishes, placed a monocular-depth text condition next to PAVR (Tab. 2). Those checkpoints are *not* the matched runs of Tab. 1 and should not be subtracted from it. In particular, the 60.1 kcal Qwen3-VL-4B Cond4 value below is an earlier-generation checkpoint, whereas 62.73 kcal in Tab. 1 is the later matched pure-base checkpoint.

Within this older table the two geometric prompts are close: Qwen3-VL-4B differs by 0.4 kcal, Qwen2-VL-2B by 1.3 kcal. That is the sense in which we say PAVR is *comparable* to monocular-depth prompting. It is not a statistical demonstration that PAVR dominates CalorieVoL.

Table 3: Depth-text ablation with a frozen Qwen3-VL-4B Cond3 adapter ($n = 346$, parse 346/346). Shuffle changes 87.86% of dish-level predictions, but MAE differences have bootstrap CIs that include zero.

Text variant	kcal MAE	Mass MAE (g)	Parse
Original	59.72	34.50	346/346
Shuffled	59.73	34.55	346/346
Constant	60.30	34.35	346/346
Removed	59.70	34.89	346/346

One further diagnostic cautions against treating monocular metric depth as a drop-in LiDAR substitute. A quality gate on reconstructed plate/food geometry passed for 86.7% of 2,033 RealSense candidates and 87.1% of relative-depth candidates, but only 12.0% of candidates from the metric variant of Depth Anything V2 (Metric-Hypersim-Small). Forced pairwise evaluations on the surviving subsets ($n = 294$ relative, $n = 41$ metric) produced calorie MAE differences whose 95% bootstrap intervals included zero. This is an exploratory diagnostic, not a third test split.

4.6 Does the Model Read the Numbers? (Cond3 probe)

If the MLLM ignored the prompt digits, shuffling them would leave predictions unchanged. We froze a Qwen3-VL-4B Cond3 adapter and replaced only the depth-derived text at test time (Tab. 3). This probe uses the Cond3 adapter, not the PAVR adapter (Cond4); the same test on Cond4 has not been run, so the results below are only an analogy for PAVR.

Predictions move: shuffling changes 87.86% of dish outputs. Aggregate MAE does not move by a robust amount; 10,000-replicate paired bootstrap 95% intervals on per-dish absolute-error differences include zero. The honest reading is therefore mixed. The model is not blind to the digits, yet on this overhead split the digits are not a large, stable contributor to *mean* calorie error once RGB LoRA is in place. That is consistent with Tab. 1, where 4B RGB already encodes much of the portion variation that is visible in Nutrition5k's relatively controlled capture geometry. Whether the PAVR descriptors behave the same way remains to be tested.

4.7 Macronutrients and a Numeric Head

We train Qwen3-VL-4B Cond4 with three seeds (42/43/44) on corrected protein/fat/carbohydrate targets and a canonical split that drops two zero-calorie training records (1,421 / 264 / 346; the test dishes are unchanged, unlike the 1,423 training dishes of Sec. 4.1); Tab. 4 gives mean ± standard deviation.

Token-level Cond4 (60.85 ± 2.06 kcal) beats a numeric-regression head that predicts the five values from the last prompt hidden state, with a train-only

Table 4: Three-seed results of Qwen3-VL-4B Cond4 on the frozen test set ($n = 346$): mean ± sample standard deviation over seeds 42/43/44. "Numeric" = regression head with ingredient auxiliary loss. MAE in kcal (energy) or g; PMAE in %.

Head	kcal MAE	kcal PMAE	Mass	Protein	Fat	Carb.
Token (deployed)	**60.85** ± 2.06	**23.35** ± 0.79	**36.14** ± 0.36	**6.17** ± 0.19	**4.76** ± 0.07	**5.54** ± 0.10
Numeric	63.34 ± 6.15	24.30 ± 2.36	41.51 ± 3.99	6.24 ± 0.77	4.91 ± 0.22	5.81 ± 0.53

Table 5: Quantization of an archived Qwen3-VL-4B Cond4 checkpoint (not the matched run in Tab. 1).

Precision	kcal MAE	Parse
bf16 (GPU)	60.14	346/346
8-bit (MLX)	60.05	346/346
4-bit (MLX)	64.7	346/346
4-bit + adapter	65.1	346/346

204-class ingredient auxiliary (63.34 ± 6.15 kcal), on every target, and is far less seed-sensitive; with three seeds the kcal difference is not established (paired difference +2.49 kcal, 95% t interval $[-17.7, +22.7]$). We therefore keep autoregressive JSON as the deployed head; these runs are separate from the matched comparison in Tab. 1.

4.8 Quantization

On-device inference uses Apple MLX [6]. For the archived Qwen3-VL-4B Cond4 checkpoint (bf16 calorie MAE 60.14, distinct from the matched 62.73 in Tab. 1), a single 8-bit quantization pass yields 60.05 kcal MAE and 4-bit yields 64.7 kcal, all with parse rate 346/346 (Tab. 5). Eight-bit is effectively lossless for calories; 4-bit costs about 4.6 kcal on this checkpoint.

4.9 Reference Comparison with Published Nutrition5k Results

Table 6 places our matched results next to published calorie MAE on Nutrition5k, with the backbone size beside each number. **These numbers are not head-to-head**: the evaluation sets differ (see the caption), our 346-dish test set comes from a filtered cohort and a random, not scan-disjoint, split (Sec. 4.1) and is likely easier, and published prompts and retry rules differ.

In kcal MAE, the 2B–4B models with PAVR (61.0–68.8 kcal) fall in the range of the 7B–13B fine-tuned MLLMs in the literature (64.3–74.2 kcal) at a fraction of the parameters, whereas zero-shot proprietary models stay at roughly 79–84 kcal even with a volume estimate (cf. the Cond1–Cond2 gap in Sec. 4.4). The original depth-aware regressors (47.6 and 41.3 kcal) are lower than every MLLM-based row, ours included (Sec. 6).

Table 6: Calorie MAE (kcal) and PMAE (%) on Nutrition5k: published systems versus ours. Other systems' rows are copied from the cited papers, *not* evaluated on our test set. Set: A = official Nutri-Test split [22] († only dishes with overhead RGB-D); B = 3,265 overhead images, 2,759/506 train/test [20]; Q = quoted in [20], split not restated; D = ours, 346 dishes (Sec. 4.1). ‡ zero-shot / with injected volume. PMAE = MAE as a percentage of the test-set mean calories (ours: MAE/260.6); "–" = not reported (the MLLM papers give MAPE, not comparable).

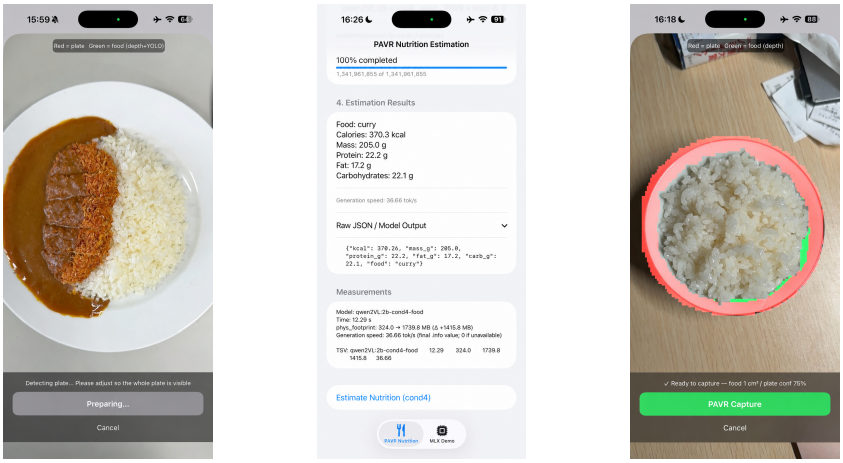
Method	Input	Backbone	Set	kcal MAE	PMAE (%)
Original Nutrition5k regressors [22]					
Predict the mean	–	–	A	150.8	60.2
2D direct	RGB	InceptionV2	A	70.6	26.1
Depth as 4th ch.	RGB-D	InceptionV2	A†	47.6	18.8
RGB + volume scalar	RGB-D	InceptionV2/V3	A†	41.3	16.5
Published MLLM-based estimators (values from [20])					
FoodLMM FT [25]	RGB	LISA 7B	Q	67.3	–
LLaVA-1.5 LoRA, 7B	RGB	7B	B	74.2	–
LLaVA-1.5 LoRA, 13B	RGB	13B	B	64.3	–
GPT-4V‡	RGB / +text	undisclosed	B	80.7 / 83.8	–
GPT-4o‡	RGB / +text	undisclosed	B	82.7 / 78.8	–
Ours (Tab. 1); Cond2 (RGB+LoRA) / Cond4 (PAVR)					
Qwen2-VL-2B	RGB / +text	2B	D	80.87 / 64.82	31.0 / 24.9
Qwen3-VL-4B	RGB / +text	4B	D	62.49 / 62.73	24.0 / 24.1
Gemma 4 E2B	RGB / +text	E2B	D	65.39 / 68.75	25.1 / 26.4
Gemma 4 E4B	RGB / +text	E4B	D	63.28 / 61.02	24.3 / 23.4

5 On-Device System

We implemented the full pipeline in an iOS app. The iPhone supplies aligned RGB and LiDAR depth; a Core ML YOLO26-seg segmenter predicts plate and food; a Swift port of the PAVR extractor emits the four descriptors; a fused 4-bit MLLM (MLX) produces JSON on device.

Figure 3 shows a capture, a completed estimate, and the plate/food overlay. The overlay in Fig. 3c is informative precisely because it is a bowl: PAVR's plane model is designed for plates, and concave ware can underestimate volume. We include the example as a limitation, not as a success case.

Preliminary timings on an iPhone 16 Pro Max, read from on-screen stage labels rather than a locked-input microbenchmark, are YOLO 1.10–1.41 s, PAVR 0.21–0.68 s, and VLM 11.40–15.25 s. The YOLO26-seg figure includes model loading in addition to mask processing and image size conversion, so it is not a pure inference time. Depth Anything V2 Small, when run alone, has a warm forward pass of 28–40 ms but a model-load cost of about 6.96 s. In the current app, YOLO, DAv2, and a 4B VLM cannot remain coresident. A fair end-to-end comparison must therefore count cold-start reloads of the depth network, not only warm milliseconds. We do not convert these ranges into a claim that PAVR



(a) Capture **(b)** Result **(c)** Plate/food overlay

Fig. 3: On-device prototype. (*a*) Live capture of a plated meal. (*b*) JSON nutrition estimate from a quantized MLLM; the macronutrient values are output by an earlier adapter and are for illustration only. (*c*) Plate (*red*) and food (*green*) overlays used by PAVR. Bowl-like ware remains a failure mode of the plate-plane assumption.

is always faster; they motivate why avoiding a second resident network matters on this device class. Older 2B measurements on a different iPhone are not mixed into these figures.

6 Discussion

What the Nutrition5k numbers can and cannot say. Nutrition5k overhead RGB-D is a favourable setting for any method that needs scale: the camera is fixed, the working distance is narrow, and a plate is usually visible. It is unsurprising that a strong 4B RGB model already estimates calories well, and that an extra geometric sentence then adds only a few percent. The 2B improvement in Tab. 1 is the more operationally relevant result, because a deployable on-device model is capacity-limited.

LiDAR versus monocular depth. If the only scoreboard is calorie MAE on this test split, monocular-depth text and PAVR are close. The case for PAVR is the *interface*: metric scale from a phone sensor, no extra vision network, and a plate plane that absorbs small tilts. Whether that interface wins in uncontrolled handheld capture is untested here. A clean next experiment is to hold the mask and the integrator fixed and swap only the depth source, and separately to hold the depth source fixed and swap a single integrated volume for the four PAVR numbers.

Relation to published systems. Because the original Nutrition5k depth-aware regressors (47.6 and 41.3 kcal) are lower than every MLLM-based row, PAVR should not be read as beating specialized RGB-D regressors on this benchmark; what it adds is a way for a small language model to use metric depth on a phone while producing free-form text output. A head-to-head evaluation of released checkpoints (e.g., FoodLMM) and of the original regressors on our 346 test dishes is the natural next step.

Limitations. PAVR assumes a locally planar plate and a single food blob. It excludes much of Nutrition5k's non-plate ware by construction. Height in Eq. (3) is an optical-axis residual. External 3D food corpora such as MetaFood3D [**?**] are not yet in the official evaluation because camera, scale, and real RGB-D provenance are incomplete; we are building a new RGB-D food dataset with real metric depth for this purpose. Our random split is not scan-disjoint (Sec. 4.1), so a scan-disjoint re-evaluation is needed to confirm the absolute numbers. Published numbers in Tab. 6 are copied from the cited papers rather than re-run.

7 Conclusion

We presented *Plate-Anchored Volumetric Reasoning* (PAVR): from a metric depth map and plate/food masks it computes four text descriptors for a multimodal LLM—the plate diameter D_{plate}, the food area A_{food}, and the 95th-percentile and mean food heights h_{95} and $\bar{h}$ above a fitted plate plane (Eq. (4))—and the LLM sees only the RGB image and these four numbers. On a frozen 346-dish Nutrition5k test split, PAVR with LoRA lowers the calorie MAE of Qwen2-VL-2B from 80.87 to 64.82 kcal (−19.9%), ties RGB-only LoRA on Qwen3-VL-4B (62.49 vs. 62.73 kcal), and is within 1.3 kcal of a monocular-depth text baseline on both Qwen models; it also runs on an iPhone with LiDAR and a 4-bit quantized MLLM. With 61.0–68.8 kcal MAE (23.4–26.4% PMAE), the 2B–4B models are in the range of published 7B–13B MLLMs (64.3–74.2 kcal) but above the original depth-aware regressors (47.6 and 41.3 kcal); test sets differ. The open question is not whether the four numbers (D_{plate}, A_{food}, h_{95}, $\bar{h}$) can replace a dedicated depth network on an overhead benchmark—they can, at similar MAE—but how far they stay reliable off the rig (distance, tilt, bowls, per-item nutrition) and on a scan-disjoint split; the number-shuffling probe has so far been run only for Cond3.

Acknowledgements This work was supported by JST CRONOS JPMJCS24K4 and JSPS KAKENHI 26K23875.

References

1. Akpa, E., Suwa, H., Arakawa, Y., Yasumoto, K.: Smartphone-based food weight and calorie estimation method for effective food journaling. SICE Journal of Control, Measurement, and System Integration **10**(5), 360–369 (2017) 3

2. Ando, Y., Ege, T., Cho, J., Yanai, K.: DepthCalorieCam: A mobile application for volume-based food calorie estimation using depth cameras. In: Proc. ACM MM Workshop on Multimedia Assisted Dietary Management (2019). https://doi.org/10.1145/3347448.3357172 1, 3
3. Fang, S., Liu, C., Zhu, F., Delp, E.J., Boushey, C.J.: Single-view food portion estimation based on geometric models. In: IEEE International Symposium on Multimedia. pp. 385–390 (2015). https://doi.org/10.1109/ISM.2015.89 1, 3
4. Fujita, H., Yanai, K.: Mobile food calorie estimation using smartphone LiDAR sensor. In: Asian Conference on Pattern Recognition. LNCS, vol. 16175, pp. 134–148. Springer (2026). https://doi.org/10.1007/978-981-95-4398-4_10 1, 2, 3
5. Han, Y., Cheng, Q., Wu, W., Huang, Z.: DPF-Nutrition: Food nutrition estimation via depth prediction and fusion. Foods **12**(23), 4293 (2023). https://doi.org/10.3390/foods12234293 3
6. Hannun, A., Digani, J., Katharopoulos, A., Collobert, R.: MLX: Efficient and flexible machine learning on Apple silicon (2023), https://github.com/ml-explore/mlx 11
7. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022) 2, 4, 7
8. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023) 4
9. Matsuda, Y., Hoashi, H., Yanai, K.: Recognition of multiple-food images by detecting candidate regions. In: IEEE International Conference on Multimedia and Expo (2012). https://doi.org/10.1109/ICME.2012.157 3
10. Myers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J., Murphy, K.: Im2Calories: Towards an automated mobile vision food diary. In: International Conference on Computer Vision. pp. 1233–1241 (2015). https://doi.org/10.1109/ICCV.2015.146 1, 3
11. Okamoto, K., Yanai, K.: An automatic calorie estimation system of food images on a smartphone. In: Proc. ACM MM Workshop on Multimedia Assisted Dietary Management (2016). https://doi.org/10.1145/2986035.2986040 3
12. Okamoto, K., Yanai, K.: UEC-FoodPIX Complete: A large-scale food image segmentation dataset. In: Proc. ICPR Workshop on Multimedia Assisted Dietary Management. pp. 647–659 (2021) 3, 4
13. Qwen Team: Qwen3-VL technical report. arXiv preprint arXiv:2511.21631 (2025) 4
14. Shao, W., Hou, S., Jia, W., Zheng, Y.: Rapid non-destructive analysis of food nutrient content using Swin-Nutrition. Foods **11**(21), 3429 (2022) 3
15. Shao, W., Min, W., Hou, S., Luo, M., Li, T., Zheng, Y., Jiang, S.: Vision-based food nutrition estimation via RGB-D fusion network. Food Chemistry **424**, 136309 (2023). https://doi.org/10.1016/j.foodchem.2023.136309 3
16. Shao, Z., Vinod, G., He, J., Zhu, F.: An end-to-end food portion estimation framework based on shape reconstruction from monocular image. In: IEEE International Conference on Multimedia and Expo. pp. 942–947 (2023) 3
17. Tai, C., Keller, M., Nair, S., Chen, Y., Wu, Y., Markham, O., Parmar, K., Xi, P., Keller, H., Kirkpatrick, S., Wong, A.: NutritionVerse: Empirical study of various dietary intake estimation approaches. arXiv preprint arXiv:2309.07704 (2023) 3
18. Tanabe, H., Yanai, K.: CalorieLLaVA: Image-based calorie estimation with multimodal large language models. In: Pattern Recognition. ICPR 2024 Interna-

tional Workshops and Challenges. LNCS, vol. 15617. Springer (2025). https://doi.org/10.1007/978-3-031-88217-3_5 4, 8
19. Tanabe, H., Yanai, K.: CalorieVoL: Integrating volumetric context into multimodal large language models for image-based calorie estimation. In: International Conference on Multimedia Modeling. LNCS, vol. 15523, pp. 353–365. Springer (2025). https://doi.org/10.1007/978-981-96-2071-5_26 2, 3, 4, 8
20. Tanabe, H., Yanai, K.: Reasoning-driven food energy estimation via multimodal large language models. Nutrients **17**(7), 1128 (2025). https://doi.org/10.3390/nu17071128 2, 4, 12
21. Tanno, R., Ege, T., Yanai, K.: AR DeepCalorieCam V2: Food calorie estimation with CNN and AR-based actual size estimation. In: Proc. ACM Symposium on Virtual Reality Software and Technology (2018). https://doi.org/10.1145/3281505.3281565 1, 3
22. Thames, Q., Karpur, A., Norris, W., Xia, F., Panait, L., Weyand, T., Sim, J.: Nutrition5k: Towards automatic nutritional understanding of generic food. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 8903–8911 (2021). https://doi.org/10.1109/CVPR46437.2021.00879 1, 2, 3, 7, 8, 12
23. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., Lin, J.: Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024) 4
24. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything V2. In: Advances in Neural Information Processing Systems (2024) 4, 8
25. Yin, Y., Qi, H., Zhu, B., Chen, J., Jiang, Y.G., Ngo, C.W.: FoodLMM: A versatile food assistant using large multi-modal model. IEEE Transactions on Multimedia **27**, 6949–6961 (2025). https://doi.org/10.1109/TMM.2025.3590924 2, 4, 12