LATENTNUTRI: ANCHOR-DRIVEN LATENT COMPUTATION FOR EFFICIENT FOOD NUTRITION ESTIMATION IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Image-based food nutrition estimation requires open-domain visual understanding and reliable continuous prediction. Existing systems typically rely on CNN/ViT regressors, which often generalize poorly beyond their training distribution, or on vision-language models (VLMs), which predict nutrition values through slow autoregressive text decoding. We introduce **LatentNutri**, an Anchor-Driven Latent Computation framework for VLM-based food nutrition estimation. LatentNutri uses Single-Pass Latent Forward to compute target-specific anchor states in one VLM forward pass, and Head-Aware Latent Regression to map these states to continuous nutrition values with lightweight regression heads. This design predicts a structured nutrition vector without autoregressive decoding. Training combines an autoregressive SFT warm-up with Scale-aware Physics-guided Regression, which jointly optimizes scale-normalized numerical accuracy and physical plausibility. Experiments on Nutrition5K and DiningBench show that LatentNutri improves prediction quality over CNN/ViT and autoregressive VLM baselines while achieving a 14.9$\times$ inference speedup over autoregressive decoding. Code is provided in the supplementary material.

1 INTRODUCTION

Image-based food nutrition estimation is an important step toward scalable dietary assessment and personalized health applications. Unlike ordinary food classification, the task requires a model to infer not only the dish category, but also portion size, ingredients, cooking style, and their combined nutritional values. These factors vary substantially across cuisines, restaurants, and image conditions, making nutrition estimation a demanding open-domain visual understanding problem with continuous outputs.

Prior work has followed two main paradigms. The first builds supervised visual regressors, often based on CNNs, ViTs, RGB-D fusion, depth estimation, or ingredient-aware modules, to map food images to calories and macronutrients (Thames et al., 2021; Shao et al., 2023; Han et al., 2023; Feng et al., 2024; Yu et al., 2025; Hou et al., 2025). These methods are efficient and can achieve strong performance on in-domain benchmarks. However, fitting the training distribution does not necessarily yield robust nutrition estimation across domains. Changes in cuisine, ingredient composition, portion presentation, and imaging conditions can weaken the visual-to-nutrition correlations learned during training, limiting their generalization to diverse real-world food images. The second paradigm uses VLMs to answer nutrition questions through natural language (Lo et al., 2024; Yin et al., 2025; Tanabe & Yanai, 2024; 2025; Yan et al., 2025). Large-scale vision-language pretraining offers broader food knowledge and visual-semantic priors, providing a promising foundation for cross-domain nutrition estimation. However, their numerical predictions are still exposed through token-by-token text generation. This text-generation interface is slow, can fail to follow the required format, and optimizes next-token likelihood rather than the continuous nutrition error that the task ultimately cares about.

We argue that the bottleneck is not the visual knowledge of VLMs, but how this knowledge is converted into continuous nutrition quantities. Rather than treating a VLM as a fixed visual encoder

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.

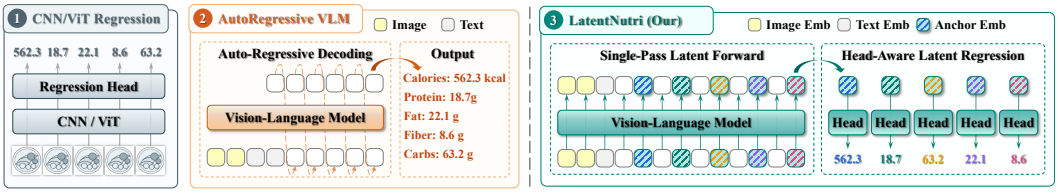


Figure 1: Prediction interfaces for image-based nutrition estimation. CNN/ViT regressors directly map visual features to nutrition values, autoregressive VLMs generate values as discrete tokens, and **LatentNutri** decodes continuous predictions from target-specific anchor states.

or asking it to verbalize nutrition values, we adapt the model so that anchor tokens induce target-specific computation in latent space.

We propose **LatentNutri**, an Anchor-Driven Latent Computation framework for VLM-based food nutrition estimation. The model receives a food image and an instruction template containing anchor tokens for calories, mass, fat, carbohydrates, and protein. Through Single-Pass Latent Forward, these anchors form contextual states specialized for each nutrition target by attending to the image and template context in one VLM pass. Head-Aware Latent Regression then decodes these computed states into scalar nutrition values with target-specific heads. The model therefore predicts a structured nutrition vector without autoregressive decoding. Figure 1 contrasts this prediction pathway with conventional visual regression and autoregressive VLM prediction.

Training the Anchor-Driven Latent Computation framework is not a standard supervised fine-tuning problem. We first use ordinary autoregressive SFT to align the model with the nutrition estimation task and output format. We then switch to Anchor-Driven Latent Computation training, where the VLM backbone and target-specific regression heads are jointly optimized using a Scale-aware Physics-guided Regression (SPG) objective. The objective combines SNAE with soft regularizers for energy consistency, mass feasibility, and non-negativity. At inference time, the default prediction requires one forward pass, while an optional test-time latent refinement loop iteratively feeds refined anchor states back into the VLM for additional latent computation.

Our contributions are threefold:

I. We introduce **LatentNutri**, an anchor-driven VLM framework that maps target-specific anchor states to continuous nutrition values in a single forward pass without autoregressive decoding.

II. We develop a progressive training strategy that transitions from SFT warm-up to latent regression, with a scale-aware, physics-guided objective for numerical accuracy and physical plausibility.

III. We construct NutriFood-75K, a large-scale dataset for food nutrition estimation. We also provide a systematic evaluation on Nutrition5K and DiningBench, demonstrating improved prediction quality and substantial efficiency gains over autoregressive VLM inference.

2 Related Work

CNN/ViT-based Food Nutrition Estimation. CNN/ViT-based approaches formulate nutrition estimation as supervised visual regression, enabled by food benchmarks with RGB images, depth, ingredients, and nutrition labels, as well as large-scale food recognition and multimodal food resources (Thames et al., 2021; Min et al., 2020; Dong et al., 2025). Recent methods improve the visual predictor with geometric, ingredient, frequency, temporal, or foundation-model features before regressing nutrition values (Han et al., 2023; Shao et al., 2023; Feng et al., 2024; Yu et al., 2025; Hou et al., 2025; Emelianov & Martinel, 2025; Yu et al., 2026; Yue et al., 2026). Despite these advances, robust cross-domain generalization remains a challenge: strong performance on a specific benchmark does not necessarily transfer to unfamiliar cuisines, portion presentations, or imaging conditions.

Autoregressive VLMs for Nutrition Estimation. VLMs offer stronger open-domain food understanding and have been used for dietary assessment through prompting, instruction tuning, and food-specific adaptation (Lo et al., 2024; Yin et al., 2025; Tanabe & Yanai, 2024; 2025; Yan et al., 2025; Khlaisamniang et al., 2025; Ma et al., 2023; Yao et al., 2024; Zhang et al., 2025; Vinod et al.,

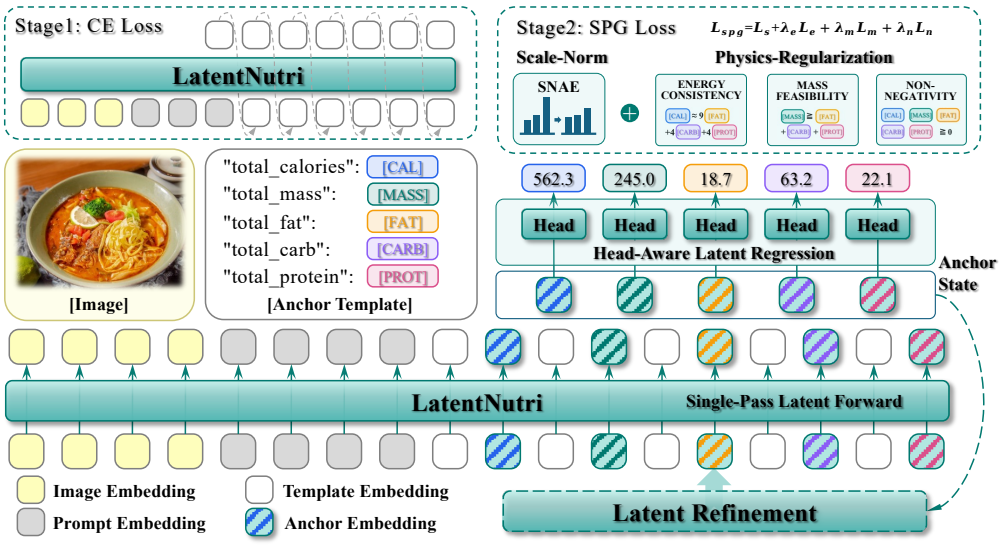


Figure 2: Overview of **LatentNutri**. Target-specific anchor tokens are placed in a structured instruction template and processed with the food image in a single VLM forward pass. The resulting contextual anchor states are decoded by parallel regression heads. Training combines SFT warm-up with SPG-guided Anchor-Driven Latent Computation training, and optional latent refinement adds test-time computation without autoregressive decoding.

2026). Recent evaluations nevertheless show that VLM-based dietary assessment remains unstable for structured numerical prediction and nutrition consistency (Jin et al., 2026; Coburn et al., 2025). FoodLMM is especially related because it introduces task-specific tokens and heads, but it still decodes them within an autoregressive generation path (Yin et al., 2025). In contrast, **LatentNutri** replaces text decoding with anchor-driven latent computation and direct continuous regression.

Latent Space Computation. A parallel line of work studies hidden states, pause tokens, or continuous thought tokens as computation carriers beyond discrete language tokens (Goyal et al., 2024; Hao et al., 2024; Lai et al., 2024; Xu et al., 2025; Shen et al., 2025; Zhang et al., 2026). Related multimodal work extends this idea to latent visual tokens and latent visual thoughts for visual reasoning (Yang et al., 2026; Wu et al., 2026; Wang et al., 2026). These works mainly target reasoning traces or text generation, whereas **LatentNutri** turns nutrition-specific anchor tokens into target-wise latent computation sites in a pretrained VLM and decodes their contextual states with regression heads, bypassing autoregressive numeric decoding.

3 Methodology

Figure 2 provides an overview of **LatentNutri**. Rather than generating nutrition values token by token, the model inserts nutrition-specific anchor tokens into a structured instruction template, lets a pretrained VLM compute their contextual states in latent space, and decodes these states with target-specific regression heads. This realizes anchor-driven continuous prediction through Single-Pass Latent Forward and Head-Aware Latent Regression.

3.1 Problem Formulation

Given a food image I, the task is to estimate the nutrition values of the main visible dish. We represent the target as

$$\mathbf{y} = [y_{\text{cal}}, y_{\text{mass}}, y_{\text{fat}}, y_{\text{carb}}, y_{\text{prot}}] \in \mathbb{R}_+^5, \tag{1}$$

where the dimensions denote total calories, total mass, fat, carbohydrates, and protein. Autoregressive VLMs cast this task as text generation and recover values by parsing the decoded answer. Although convenient, this text-generation interface optimizes next-token likelihood rather than continuous nutrition error, and inference depends on both the length and the format of the generated

Instruction Template for LatentNutri

USER: `<image>`

Look at this food dish image and estimate the nutrition values. Return a single JSON object with exactly these keys: `total_calories`, `total_mass`, `total_fat`, `total_carb`, `total_protein`. Output special placeholder tokens as values: `[CAL]`, `[MASS]`, `[FAT]`, `[CARB]`, `[PROT]`. Only output valid JSON.

ASSISTANT: `{"total_calories": [CAL], "total_mass": [MASS], "total_fat": [FAT], "total_carb": [CARB], "total_protein": [PROT]}`

*Hidden states at **bold** anchor positions serve as latent nutrition states for regression.*

Figure 3: Instruction template for **LatentNutri**. The user and assistant turns are processed in one VLM forward pass.

string. **LatentNutri** instead learns

$$F_\phi(I, x_\mathcal{A}) \to \hat{\mathbf{y}}, \tag{2}$$

where $x_\mathcal{A}$ is an instruction template containing anchor tokens and $\hat{\mathbf{y}}$ is predicted in the original continuous label space. The goal is therefore to retain VLM conditioning while aligning training and inference with the numerical structure of the task.

3.2 Anchor-Driven Latent Computation

Let f_θ denote the VLM. We introduce five anchor tokens,

$$\mathcal{A} = \{a_{\text{cal}}, a_{\text{mass}}, a_{\text{fat}}, a_{\text{carb}}, a_{\text{prot}}\}, \tag{3}$$

as special tokens with trainable input embeddings $\{\mathbf{e}_j\}_{a_j \in \mathcal{A}}$. Anchor-Driven Latent Computation uses these anchors as target-specific positions where the VLM computes nutrition-relevant representations.

Single-Pass Latent Forward. The template asks the model to estimate total calories at `[CAL]`, total mass at `[MASS]`, fat at `[FAT]`, carbohydrates at `[CARB]`, and protein at `[PROT]` (Figure 3). The anchors are not appended as a separate block. They appear inside field-specific contexts, surrounded by attribute names, punctuation, and short task instructions. Thus each readout position receives both visual evidence from the image and textual context identifying the target variable. The user instruction and the assistant template are concatenated and processed in one VLM forward pass. This one-pass computation is also the first round in the optional refinement procedure introduced below.

After this forward pass, we collect the final-layer hidden states:

$$\mathbf{H}^{(1)} = f_\theta(I, x_\mathcal{A}) \in \mathbb{R}^{T \times d}. \tag{4}$$

Here T denotes the multimodal input sequence length and d is the hidden-state dimension of the VLM. For each anchor token a_j, let p_j denote its position in the assistant template within the full multimodal input sequence. The contextual anchor state is read from this position:

$$\mathbf{h}_j^{(1)} = \mathbf{H}_{p_j}^{(1)}. \tag{5}$$

When no refinement is used, we write $\mathbf{h}_j$ for $\mathbf{h}_j^{(1)}$ for brevity. The anchor token specifies where the model should form the target-specific representation. The latent nutrition value is represented by the contextual state produced after multimodal attention over the image and instruction.

Head-Aware Latent Regression. Let $\mathcal{J} = \{\text{cal}, \text{mass}, \text{fat}, \text{carb}, \text{prot}\}$ denote the set of nutrition targets. Each contextual anchor state is decoded by a lightweight regression head:

$$\hat{y}_j = g_j(\mathbf{h}_j), \quad j \in \mathcal{J}. \tag{6}$$

In the default architecture, g_j is a two-layer MLP:

$$g_j(\mathbf{h}_j) = \mathbf{w}_{j,2}^\top \sigma(\mathbf{W}_{j,1}\,\text{LN}(\mathbf{h}_j) + \mathbf{b}_{j,1}) + b_{j,2}, \tag{7}$$

where LN denotes layer normalization and σ is a nonlinear activation. This design is head-aware because calories, mass, and macronutrients differ in scale and visual evidence. All heads run in parallel after the shared VLM forward pass.

The complete prediction is

$$\hat{\mathbf{y}} = [\hat{y}_{\text{cal}}, \hat{y}_{\text{mass}}, \hat{y}_{\text{fat}}, \hat{y}_{\text{carb}}, \hat{y}_{\text{prot}}]. \tag{8}$$

3.3 Progressive Training Strategy

We train **LatentNutri** with a progressive strategy that starts from SFT warm-up and then transitions to Anchor-Driven Latent Computation training. The first stage adapts the VLM to the nutrition estimation task through ordinary autoregressive supervised fine-tuning. Given a food image and a prompt instruction, the model generates a structured text answer with standard cross-entropy loss:

$$\mathcal{L}_{\text{sft}} = -\sum_{t=1}^{M} \log p_\theta(z_t \mid z_{<t}, I, x_{\text{user}}), \tag{9}$$

where x_{user} denotes the ordinary user instruction and z_t is the target response token. This stage is used only as a warm-up. It aligns the backbone with nutrition-related visual attributes and structured nutrition fields before the prediction pathway is changed.

The second stage directly optimizes the Anchor-Driven Latent Computation framework. Given the image and anchor template, Single-Pass Latent Forward forms contextual anchor states. Head-Aware Latent Regression then maps these states to $\hat{\mathbf{y}}$. We jointly update the VLM and the heads, so the backbone learns to shape the anchor states for continuous prediction.

Scale-normalized absolute error. Nutrition dimensions have very different numerical ranges. Calories and mass can be orders of magnitude larger than individual macronutrients, so an unnormalized absolute loss would be dominated by large-scale targets. We therefore use Scale-Normalized Absolute Error (SNAE) as the main regression term:

$$\mathcal{L}_{\text{snae}} = \frac{1}{Z} \sum_{j \in \mathcal{J}} \alpha_j \frac{|\hat{y}_j - y_j|}{|y_j| + \epsilon}. \tag{10}$$

Here α_j controls the relative weight of each nutrition dimension, $Z = \sum_{j \in \mathcal{J}} \alpha_j$, and ϵ prevents instability for very small targets. This term balances supervision across calories, mass, and macronutrients.

Physics-guided regularization. Food nutrition labels are noisy because recipes, portions , and hidden ingredients rarely have a single exact ground truth. We therefore add soft regularizers derived from standard food-composition accounting: Atwater general factors relate metabolizable energy to macronutrient mass, and mass-balance feasibility requires the aggregate macronutrient mass not to exceed total food mass(Maclean et al., 2003).

The energy implied by macronutrients is

$$\hat{e}_{\text{macro}} \triangleq 9\hat{y}_{\text{fat}} + 4\hat{y}_{\text{carb}} + 4\hat{y}_{\text{prot}}, \tag{11}$$

where fat contributes roughly 9 kcal/g and carbohydrates and protein contribute roughly 4 kcal/g. Because real dishes may include fiber, alcohol, water, sauces, and imperfect labels, the energy term uses a relative margin:

$$\mathcal{L}_{\text{energy}} = \left[|\hat{y}_{\text{cal}} - \hat{e}_{\text{macro}}| - \rho|\,\text{sg}(\hat{y}_{\text{cal}})|\right]_+. \tag{12}$$

Here $[\cdot]_+$ is the ReLU operator, ρ is the margin ratio, and $\text{sg}(\cdot)$ stops gradients through the margin scale. The hinge form penalizes clear violations without forcing exact equality. The mass feasibility term penalizes predicted macronutrients that exceed total mass:

$$\mathcal{L}_{\text{mass}} = \left[\hat{y}_{\text{fat}} + \hat{y}_{\text{carb}} + \hat{y}_{\text{prot}} - \hat{y}_{\text{mass}}\right]_+. \tag{13}$$

Finally, non-negativity discourages invalid negative values:

$$\mathcal{L}_{\text{nonneg}} = \sum_{j \in \mathcal{J}} [-\hat{y}_j]_+. \tag{14}$$

The three regularizers form the physics-guided term, which is combined with SNAE to obtain the full Scale-aware Physics-guided Regression (SPG) objective:

$$\begin{aligned} \mathcal{L}_{\text{phys}} &= \lambda_e \mathcal{L}_{\text{energy}} + \lambda_m \mathcal{L}_{\text{mass}} + \lambda_n \mathcal{L}_{\text{nonneg}}, \\ \mathcal{L}_{\text{spg}} &= \mathcal{L}_{\text{snae}} + \mathcal{L}_{\text{phys}}. \end{aligned} \tag{15}$$

SNAE normalizes each absolute error by the target scale, preventing large-magnitude targets such as calories and mass from dominating the gradients. The physics terms are implemented as hinge penalties: they contribute no loss when energy consistency, mass feasibility, and non-negativity are satisfied, and provide corrective gradients only when these constraints are violated. This soft-constraint design is suitable for food images, where exact recipes, portions, and hidden ingredients are often unobserved.

3.4 Test-time Latent Refinement

The default inference mode stops after the first round. When additional computation is acceptable, later rounds reuse the same image and text template but replace only the embeddings at the five assistant-template positions p_j with transformed states from the previous round. For round $k \geq 2$, a learnable projection r_ψ (Appendix A.4) combines the previous anchor state with the learned embedding of the same anchor token:

$$\tilde{\mathbf{e}}_j^{(k)} = r_\psi(\mathbf{h}_j^{(k-1)}, \mathbf{e}_j), \quad a_j \in \mathcal{A}. \tag{16}$$

Let $x_{\mathcal{A}}^{(k)}$ denote the same template as $x_{\mathcal{A}}$, except that the learned embedding $\mathbf{e}_j$ at each position p_j is replaced by $\tilde{\mathbf{e}}_j^{(k)}$. All ordinary text tokens and image inputs remain unchanged. The current round then produces updated anchor states:

$$\mathbf{H}^{(k)} = f_\theta(I, x_{\mathcal{A}}^{(k)}), \quad \mathbf{h}_j^{(k)} = \mathbf{H}_{p_j}^{(k)}, \quad a_j \in \mathcal{A}. \tag{17}$$

The regression heads are applied to the final refined states:

$$\hat{\mathbf{y}}^{(K)} = [g_j(\mathbf{h}_j^{(K)})]_{j \in \mathcal{J}}. \tag{18}$$

During refinement training, we unroll the loop for the same number of rounds K used at inference and optimize the following objective:

$$\mathcal{L}_{\text{ref}} = \sum_{k=1}^{K} \omega_k \mathcal{L}_{\text{snae}}(\hat{\mathbf{y}}^{(k)}, \mathbf{y}) + \mathcal{L}_{\text{phys}}(\hat{\mathbf{y}}^{(K)}), \tag{19}$$

where ω_k are round weights. The loop operates entirely in latent space. It does not generate intermediate rationales or increase the number of decoded text tokens.

3.5 NutriFood-75K Construction

Large-scale, reliable per-image nutrition annotations remain limited for open-domain food images. To provide scalable supervision, we construct NutriFood-75K from MM-Food-100K (Dong et al., 2025) and ISIA Food-500 (Min et al., 2020) using a two-stage VLM-assisted pipeline. Qwen3-VL-8B-Instruct first filters visually unreliable images. Gemini-3.1-Pro then reviews the retained images and estimates calories, mass, fat, carbohydrates, and protein. After removing malformed, low-confidence, and physically implausible annotations, NutriFood-75K contains 75,547 training samples. These include 26,564 samples from MM-Food-100K and 48,983 from ISIA Food-500. Full construction details and prompts are provided in Appendix C.1 and Appendix C.3.

4 Experiments

4.1 Experimental Setup

Implementation details. **LatentNutri** uses Qwen3-VL-4B as the default VLM backbone. Unless otherwise stated, **LatentNutri** denotes the Single-Pass Latent Forward model with $K = 1$ at inference time. We train all supervised models on the same training pool consisting of NutriFood-75K and the Nutrition5K train split, and evaluate on the Nutrition5K test split and DiningBench, where DiningBench serves as an out-of-distribution benchmark for open-domain restaurant food images. The SFT warm-up runs for 1 epoch and the Anchor-Driven Latent Computation stage runs for 3 epochs. Full hyperparameters are listed in Appendix A.

Method	Calories		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
CNN/ViT baselines										
ConvNeXt-B	226.4	29.8	14.31	38.8	25.76	45.3	15.57	36.6	70.51	37.60
Swin-B	232.1	30.6	14.68	39.8	25.34	44.5	16.82	39.6	72.22	38.60
Google-Nutrition-rgb	255.9	33.7	15.39	41.7	29.62	52.0	17.51	41.2	79.62	42.20
Zero-shot VLMs										
Gemini-3-Pro	175.2	23.1	**10.42**	**28.2**	**13.95**	**24.5**	**10.83**	**25.5**	52.60	**25.30**
GPT-4.1	312.4	41.1	14.24	38.6	17.93	31.5	23.48	55.3	92.00	41.61
Qwen3-VL-4B	353.6	46.6	14.27	38.6	20.29	35.7	19.21	45.2	101.83	41.52
Qwen3-VL-8B	264.0	34.8	14.00	37.9	19.41	34.1	19.20	45.2	79.16	37.99
InternVL3.5-30B-A3B	332.9	43.8	16.49	44.7	25.24	44.3	22.53	53.0	99.28	46.46
SFT VLMs										
Qwen3-VL-4B-SFT	185.7	24.4	11.67	31.6	16.87	29.6	13.12	30.9	56.83	29.14
Qwen3-VL-8B-SFT	183.2	24.1	11.23	30.4	16.02	28.1	12.96	30.5	55.85	28.29
FoodLMM	205.0	27.0	11.78	31.9	21.11	37.1	14.30	33.7	63.05	32.41
Ours										
LatentNutri	**171.3**	**22.6**	10.85	29.4	15.13	26.6	11.98	28.2	**52.32**	26.68

Table 1: Representative results on DiningBench. MAE and PMAE are reported for each nutrition target and macro-averaged across targets. Results for all 18 evaluated methods are reported in Table 15 (Appendix E.1). Best and second-best results are shown in **bold** and underlined.

Method	Calories		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
Google-Nutrition-rgb	70.6	26.1	5.50	29.5	6.10	31.9	5.00	34.2	25.52	28.10
Gemini-3-Pro	71.8	28.1	5.58	31.9	6.42	32.6	5.56	43.3	27.85	32.20
Qwen3-VL-8B-SFT	49.6	19.4	**4.16**	**23.8**	4.58	23.3	3.81	29.7	18.58	22.33
LatentNutri	**46.6**	**18.3**	4.33	24.8	**4.28**	**21.7**	**3.62**	**28.2**	**17.48**	**21.47**

Table 2: Representative Nutrition5K results. Avg. is computed over all five targets, including mass. Results for all 18 evaluated methods are reported in Table 16 (Appendix E.2). Best and second-best results are shown in **bold** and underlined.

Baseline methods. We compare with three groups of baselines: (1) supervised CNN/ViT regressors trained on the same data with standard regression objectives, (2) zero-shot VLMs, covering both closed-source and open-source models, and (3) autoregressive SFT VLMs trained on the same data. Detailed descriptions of all baseline models are provided in Appendix A.

4.2 Main Results

Table 1 summarizes representative results on DiningBench, and Table 2 gives the representative companion summary on Nutrition5K. We use average PMAE (Thames et al., 2021) as the primary comparison. On DiningBench, **LatentNutri** substantially improves over supervised visual regressors, reducing the average PMAE from 37.60% for the strongest CNN/ViT baseline to 26.68%. It also lowers the average PMAE of the strongest autoregressive SFT VLM from 28.29% to 26.68%, suggesting that training the VLM to compute continuous nutrition states in latent space better matches the task than generated text. The strongest zero-shot model, Gemini-3-Pro, remains highly competitive, while **LatentNutri** provides a strong open-source alternative with direct continuous prediction. On Nutrition5K, **LatentNutri** shows the same overall trend. In the representative summary, it reduces average PMAE from 28.1% for the best CNN/ViT baseline and 22.33% for the best SFT VLM baseline to 21.47%.

4.3 Superior Efficiency of LatentNutri

At inference time, our method is substantially more efficient because Single-Pass Latent Forward replaces token-by-token numeric decoding. Figure 4 compares latency across multiple Qwen3-VL

backbone sizes on DiningBench. For the 4B backbone, the Single-Pass Latent Forward model takes 126.4 ms per sample, while an autoregressive SFT VLM requires 1878.3 ms per sample to generate 47.3 tokens on average. This corresponds to a 14.9× speedup. The optional 3-forward latent refinement setting takes 255.2 ms and remains 7.4× faster than autoregressive decoding with the same 4B backbone.

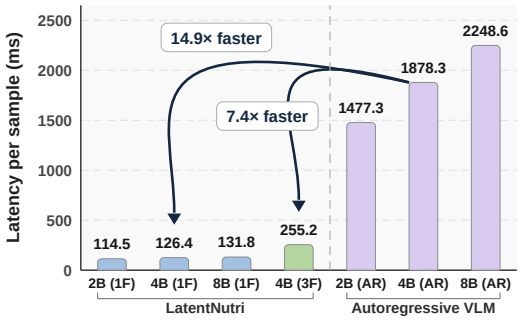


Figure 4: Inference latency on DiningBench across Qwen3-VL backbone sizes. LatentNutri achieves substantial speedups over autoregressive decoding.

4.4 Ablation Study

Training recipe and objective. The two-stage training recipe is important for stable latent computation. Removing the autoregressive SFT warm-up increases average PMAE from 26.68% to 28.07% on DiningBench, suggesting that warm-up gives the VLM useful task priors before continuous anchor-state training. Removing PhysReg also degrades both datasets, while removing SNAE causes a larger drop. This indicates that the two parts of SPG loss are complementary. Physical priors regularize implausible outputs, and scale-normalized absolute supervision remains critical for accurate nutrition regression.

Regression head design. The lightweight decoder architecture also matters. Replacing the default independent two-layer heads with one-layer heads increases average MAE, especially on DiningBench. Sharing one two-layer head across all nutrition targets performs even worse, indicating that calories, mass, and macronutrients benefit from target-specific decoding of their corresponding computed anchor states. We therefore keep separate lightweight heads.

Variant	DiningBench		Nutrition5K	
	MAE	PMAE(%)	MAE	PMAE(%)
Training recipe and objective				
w/o SNAE	56.32	32.89	19.22	24.77
w/o SFT warm-up	54.67	28.07	18.02	22.26
w/o PhysReg	52.91	27.08	17.74	21.94
Full model	**52.32**	**26.68**	**17.48**	**21.47**
Regression head design				
Shared 2-layer head	223.95	100.00	100.25	97.47
1-layer MLP heads	159.69	63.21	41.00	40.67
Default 2-layer heads	**52.32**	**26.68**	**17.48**	**21.47**
VLM backbone adaptation				
Frozen VLM	110.36	47.91	36.60	40.43
Full fine-tuning	**52.32**	**26.68**	**17.48**	**21.47**

Table 3: Ablation study of the training objective, regression heads, and backbone adaptation. Average MAE and PMAE are reported on DiningBench and Nutrition5K.

VLM backbone adaptation. To assess whether backbone fine-tuning is necessary, we compare against a frozen variant that trains only the regression heads while keeping all VLM parameters fixed. This variant more than doubles average MAE on both datasets, confirming that **LatentNutri** does not merely probe a fixed representation. Rather, the backbone must actively adapt its anchor-token states in latent space to develop nutrition-relevant internal representations that integrate visual, semantic, and quantitative signals.

4.5 In-depth Analysis

Effect of test-time latent refinement. We evaluate the optional latent refinement loop across different numbers of rounds, with matched training and inference depths for each configuration. As shown in Figure 6, additional latent computation generally improves average PMAE. On DiningBench, PMAE decreases from 26.68% to 26.04% after three rounds. On Nutrition5K, PMAE decreases more steadily from 21.47% to 20.51% over five rounds. Refinement can improve accuracy with additional computation, while Single-Pass Latent Forward remains the default for efficient inference.

Effect of VLM backbone scale. We evaluate Anchor-Driven Latent Computation with 2B, 4B, and 8B Qwen3-VL backbones, keeping the anchor template, regression heads, and training recipe fixed. Full results are reported in Appendix E.3. The results show that **LatentNutri** benefits from

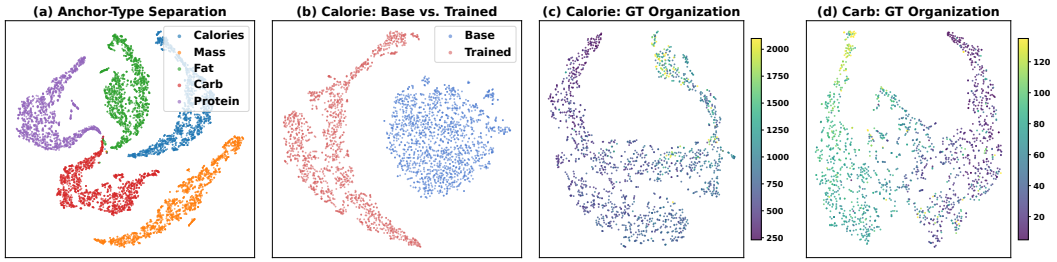


Figure 5: t-SNE visualization of anchor-token states in trained 4B **LatentNutri**. Panels show target separation, training-induced shifts, and organization by ground-truth nutrition values.

stronger VLM backbones. On DiningBench, average PMAE decreases from 27.43% to 26.68% and 26.55% as the backbone scales from 2B to 4B and 8B. This trend indicates that as the underlying VLM becomes more capable, our method can exploit richer visual-semantic knowledge and stronger latent-space computation to achieve better nutrition prediction. Together with the cross-family study on Qwen2.5-VL backbones in Appendix B.3, these results suggest that **LatentNutri** provides a general framework for adapting pretrained VLMs to numeric nutrition estimation.

Latent state analysis of semantic anchors. We analyze the output hidden states at the special anchor-token positions using t-SNE (Van der Maaten & Hinton, 2008). These states are taken before the regression heads, so the visualization reflects the latent computation produced by the VLM. Figure 5 summarizes three observations. First, the states of the five anchor tokens form clearly separated regions, indicating that they are not interchangeable placeholders. Second, the calorie anchor states move away from the pretrained distribution after training, showing that training adapts the contextual anchor states rather than only fitting the regression heads. Third, trained calorie and carbohydrate anchor states vary smoothly with ground-truth values, suggesting that the latent space preserves target-relevant numerical structure. Together, these results show that the VLM computes semantically organized, nutrition-relevant states at anchor positions to support numerical nutrition estimation. Additional analyses of physical plausibility, anchor-template design, backbone-scale representations, training cost, qualitative cases, and dataset effectiveness are provided in the appendix.

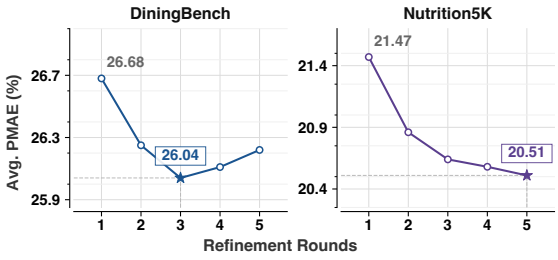


Figure 6: Effect of test-time latent refinement. Average PMAE is reported over refinement rounds.

5 Conclusion and Discussion

We introduced **LatentNutri**, an Anchor-Driven Latent Computation framework for image-based food nutrition estimation. Instead of decoding nutrition values token by token, LatentNutri computes target-specific anchor states in a pretrained VLM and maps them to continuous nutrition values with lightweight regression heads. To train this framework reliably, we combine an autoregressive SFT warm-up with SPG-guided latent computation training. Experiments on Nutrition5K and DiningBench show that this design improves prediction accuracy over CNN/ViT regressors and autoregressive VLM baselines while substantially reducing inference latency.

Although this work focuses on food nutrition estimation, we believe this framework can extend beyond this setting. **LatentNutri** suggests a way to adapt pretrained VLMs and LLMs for structured continuous prediction without autoregressive numeric generation. This perspective may be useful for domains that require numerical estimation from complex visual and textual evidence, such as clinical measurement, concentration, or dosage prediction. We hope our work can provide a useful perspective for future research in this direction.

AI USE STATEMENT

During manuscript preparation, generative AI tools were used for language polishing and presentation formatting. Separately, the NutriFood-75K construction pipeline uses Qwen3-VL-8B-Instruct and Gemini-3.1-Pro for image-quality filtering, review, and nutrition annotation, as described in Section 3.5 and Appendix C.1. The authors take responsibility for the scientific content, reported results, and final manuscript.

REPRODUCIBILITY STATEMENT

Section 3 specifies the model and training objectives, and Appendix A provides implementation details, hyperparameters, and evaluation protocols. The NutriFood-75K construction process is described in Section 3.5 and Appendix C.1, with the full prompts in Appendix C.3. All code is provided as an anonymized ZIP archive in the supplementary material. We will publicly release the constructed NutriFood-75K dataset and trained model checkpoints.

ETHICS STATEMENT

This work studies image-based nutrition estimation for research purposes. Predictions can be affected by visually unobservable ingredients, portion sizes, and annotation noise, and should not substitute for professional dietary assessment. Limitations and potential risks are discussed in Appendix G. The planned release of NutriFood-75K will respect the licenses and access conditions of its source datasets.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024.

Bruce Coburn, Jiangpeng He, Megan E Rollo, Satvinder S Dhaliwal, Deborah A Kerr, and Fengqing Zhu. Evaluating large multimodal models for nutrition analysis: A new benchmark enriched with contextual metadata. In *2025 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–7. IEEE, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Yi Dong, Yusuke Muraoka, Scott Shi, and Yi Zhang. Mm-food-100k: A 100,000-sample multimodal food intelligence dataset with verifiable provenance. *arXiv preprint arXiv:2508.10429*, 2025.

Vitalii Emelianov and Niki Martinel. Nutrition prediction from food images using foundation models. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2025.

Zhihui Feng, Hao Xiong, Weiqing Min, Sujuan Hou, Huichuan Duan, Zhonghua Liu, and Shuqiang Jiang. Ingredient-guided rgb-d fusion network for nutritional assessment. *IEEE Transactions on AgriFood Electronics*, 3(1):156–166, 2024.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *International Conference on Learning Representations*, volume 2024, pp. 27896–27923, 2024.

Yuzhe Han, Qimin Cheng, Wenjin Wu, and Ziyang Huang. Dpf-nutrition: Food nutrition estimation via depth prediction and fusion. *Foods*, 12(23):4293, 2023.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sujuan Hou, Zhihui Feng, Hao Xiong, Weiqing Min, Peng Li, and Shuqiang Jiang. Dsdgf-nutri: A decoupled self-distillation network with gating fusion for food nutritional assessment. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 5218–5227, 2025.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Song Jin, Juntian Zhang, Xun Zhang, Zeying Tian, Fei Jiang, Guojun Yin, Wei Lin, Yong Liu, and Rui Yan. Diningbench: A hierarchical multi-view benchmark for perception and reasoning in the dietary domain. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35289–35310, 2026.

Pitikorn Khlaisamniang, Kun Kerdthaisong, Supasate Vorathammathorn, Nutchanon Yongsatianchot, Hirunkul Phimsiri, Amrest Chinkamol, Teermade Thitseesaeng, Kanyakorn Veerakanjana, Kaisorn Kachai, Piyalitt Ittichaiwong, et al. Decomposing food images for better nutrition analysis: a nutritionist-inspired two-step multimodal llm approach. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 482–491. IEEE, 2025.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9579–9589. IEEE, 2024.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9992–10002. Ieee, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11966–11976. IEEE, 2022.

Frank P-W Lo, Jianing Qiu, Zeyu Wang, Junhong Chen, Bo Xiao, Wu Yuan, Stamatia Giannarou, Gary Frost, and Benny Lo. Dietary assessment with multimodal chatgpt: A systematic analysis. *IEEE Journal of Biomedical and Health Informatics*, 28(12):7577–7587, 2024.

Peihua Ma, Yixin Wu, Ning Yu, Yang Zhang, Michael Backes, Qin Wang, and Cheng-I Wei. Umdfood: Vision-language models boost food composition compilation. *arXiv preprint arXiv:2306.01747*, 2023.

William Maclean, James Harnly, Junshi Chen, Simon Chevassus-Agnes, G Gilani, Geoffrey Livesey, and P Warwick. Food energy–methods of analysis and conversion factors. In *Food and agriculture organization of the united nations technical workshop report*, volume 77. The Food and Agriculture Organization Rome, Italy, 2003.

Weiqing Min, Linhu Liu, Zhiling Wang, Zhengdong Luo, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 393–401, 2020.

Wenjing Shao, Weiqing Min, Sujuan Hou, Mengjiang Luo, Tianhao Li, Yuanjie Zheng, and Shuqiang Jiang. Vision-based food nutrition estimation via rgb-d fusion network. *Food chemistry*, 424:136309, 2023.

Xuan Shen, Yizhou Wang, Yufa Zhou, Xiangxi Shi, Pu Zhao, Yanzhi Wang, and Jiuxiang Gu. Efficient reasoning with hidden thinking. *arXiv preprint arXiv:2501.19201*, 2025.

Hikaru Tanabe and Keiji Yanai. Caloriellava: Image-based calorie estimation with multimodal large language models. In *International Conference on Pattern Recognition*, pp. 63–75. Springer, 2024.

Hikaru Tanabe and Keiji Yanai. Calorievol: Integrating volumetric context into multimodal large language models for image-based calorie estimation. In *International Conference on Multimedia Modeling*, pp. 353–365. Springer, 2025.

Quin Thames, Arjun Karpur, Wade Norris, Fangting Xia, Liviu Panait, Tobias Weyand, and Jack Sim. Nutrition5k: Towards automatic nutritional understanding of generic food. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8899–8907. IEEE, 2021.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Gautham Vinod, Siddeshwar Raghavan, Bruce Coburn, and Fengqing Zhu. Dietdelta: A vision-language approach for dietary assessment via before-and-after images. *arXiv preprint arXiv:2604.06352*, 2026.

Yubo Wang, Juntian Zhang, Yichen Wu, Yankai Lin, Nils Lukas, and Yuhan Liu. Forest before trees: Latent superposition for efficient visual reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11272–11288, 2026.

Linquan Wu, Tianxiang Jiang, Yifei Dong, Haoyu Yang, Fengji Zhang, Shichang Meng, Ai Xuan, Linqi Song, and Jacky Keung. Lavit: Aligning latent visual thoughts for multi-modal reasoning. In *European Conference on Computer Vision*, pp. 345–363. Springer, 2026.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot: Soft chain-of-thought for efficient reasoning with llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23336–23351, 2025.

Runze Yan, Hanqi Luo, Jiaying Lu, Darren Liu, Hannah Posluszny, Mehak Preet Dhaliwal, Janice MacLeod, Yao Qin, Carl Yang, Terry J Hartman, et al. Dietai24 as a framework for comprehensive nutrition estimation using multimodal large language models. *Communications Medicine*, 5(1): 458, 2025.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33510–33520, 2026.

Dongyu Yao, Keling Yao, Junhong Zhou, and Yinghao Zhang. Caloraify: Calorie estimation with visual-text pairing and lora-driven visual language models. *arXiv preprint arXiv:2412.09936*, 2024.

Yuehao Yin, Huiyan Qi, Bin Zhu, Jingjing Chen, Yu-Gang Jiang, and Chong-Wah Ngo. Foodlmm: A versatile food assistant using large multi-modal model. *IEEE Transactions on Multimedia*, 2025.

Dongjian Yu, Weiqing Min, Xin Jin, Qian Jiang, and Shuqiang Jiang. Spatial-aware multi-modal information fusion for food nutrition estimation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8863–8871, 2025.

Dongjian Yu, Weiqing Min, Qian Jiang, Xing Lin, Xin Jin, and Shuqiang Jiang. Omnifood8k: Single-image nutrition estimation via hierarchical frequency-aligned fusion. *arXiv preprint arXiv:2604.12356*, 2026.

Chengkun Yue, Chuanzhi Xu, and Jiangpeng He. V-nutri: Dish-level nutrition estimation from egocentric cooking videos. *arXiv preprint arXiv:2604.11913*, 2026.

Pengfei Zhang, Yutong Song, Chenhan Lyu, Ziyu Wang, and Amir M Rahmani. Foodagent: A multi-modal mixture of experts reasoning agent for divide-and-conquer food nutrition estimation. In *2025 IEEE 21st International Conference on Body Sensor Networks (BSN)*, pp. 1–4. IEEE, 2025.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, and Xin Wang. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *Advances in Neural Information Processing Systems*, 38:168990–169012, 2026.

A Implementation Details

A.1 Training Setup

LatentNutri uses Qwen3-VL-4B as the default VLM backbone. **LatentNutri** is trained in two stages. First, we perform an autoregressive SFT warm-up for one epoch with a batch size of 32 and a learning rate of 1×10^{-5}. This stage uses full fine-tuning of the VLM backbone. Second, we switch to Anchor-Driven Latent Computation training and train the VLM backbone and target-specific regression heads for three epochs with a batch size of 32 and a learning rate of 5×10^{-6}. All trainable parameters are optimized with AdamW, and training uses bf16 precision and DeepSpeed ZeRO-2. For the SPG objective, we set the energy margin ratio to $\rho = 0.2$ and use $\lambda_e = \lambda_m = \lambda_n = 0.001$ for the energy, mass, and non-negativity regularizers. The scale-aware regression weights are set to $\alpha_j = 1$ for all nutrition targets, and the denominator stabilizer is set to $\epsilon = 1$. For refinement training, we supervise only the final refinement output by setting $\omega_K = 1$ and $\omega_k = 0$ for earlier rounds. All experiments are run on four NVIDIA A100 80GB GPUs.

A.2 Baseline Methods

Group 1: Supervised CNN/ViT regressors. We include ResNet-101 (He et al., 2016), ConvNeXt-B (Liu et al., 2022), Swin-B (Liu et al., 2021), and the Google-Nutrition-rgb from Thames et al. (2021). All models take a single RGB image as input and are trained end-to-end with a mean squared error regression objective on the five nutrition targets (calories, protein, carbohydrate, fat, and total mass).

Group 2: Zero-shot VLMs. We cite one representative reference for each model family. Closed-source models include Gemini-2.5-Pro and Gemini-3-Pro from Google DeepMind (Comanici et al., 2025), as well as GPT-4.1 from OpenAI (Hurst et al., 2024). Open-source models include Qwen2.5-VL (3B, 7B) and Qwen3-VL (4B, 8B) from Alibaba (Bai et al., 2025), and InternVL3.5-30B-A3B from Shanghai AI Lab (Chen et al., 2024). All zero-shot models are prompted with the food image and asked to estimate the five nutrition values in a structured JSON format, with no task-specific fine-tuning. The exact prompt used for evaluation is provided in Table 21.

Group 3: Autoregressive SFT VLMs. We fine-tune Qwen2.5-VL (3B, 7B) and Qwen3-VL (2B, 4B, 8B) (Bai et al., 2025) on the same training pool for three epochs using a standard sequence-to-sequence cross-entropy objective. At inference time, nutrition values are generated as structured text and then parsed into numbers.

A.3 Evaluation Metrics

We report mean absolute error (MAE) and percentage mean absolute error (PMAE). For target j, they are defined as

$$\mathrm{MAE}_j = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_{ij} - y_{ij}|, \tag{20}$$

$$\mathrm{PMAE}_j = \frac{\mathrm{MAE}_j}{\frac{1}{N} \sum_{i=1}^{N} y_{ij}} \times 100\%. \tag{21}$$

Average scores are macro-averaged over the reported nutrition targets. DiningBench does not provide total mass annotations, so mass is excluded from DiningBench evaluation. MAE and PMAE are lower-is-better. Full experimental settings and hyperparameters are summarized in Table 4.

Category	Parameter	Value
Data	Training data	NutriFood-75K + Nutrition5K train split
Data	Evaluation data	Nutrition5K test split + DiningBench
Hardware	GPUs	4 × NVIDIA A100 80GB
Model	Default backbone	Qwen3-VL-4B
Optimization	Optimizer	AdamW
Optimization	Precision	bf16
Optimization	Distributed training	DeepSpeed ZeRO-2
SFT warm-up	Epochs	1
SFT warm-up	Batch size	32
SFT warm-up	Learning rate	1×10^{-5}
SFT warm-up	Fine-tuning mode	Full fine-tuning
Latent-computation stage	Epochs	3
Latent-computation stage	Batch size	32
Latent-computation stage	Learning rate	5×10^{-6}
Refinement	Round weights	$\omega_K = 1$ and $\omega_k = 0$ for $k < K$
SPG loss	Energy margin	$\rho = 0.2$
SPG loss	Energy weight	$\lambda_e = 0.001$
SPG loss	Mass weight	$\lambda_m = 0.001$
SPG loss	Non-negativity weight	$\lambda_n = 0.001$
SPG loss	Target weights	$\alpha_j = 1$ for all targets
SPG loss	Stabilizer	$\epsilon = 1$
SFT VLM baselines	Training schedule	3 epochs on the same training data
CNN/ViT baselines	Training data	Same training data with supervised regression objectives

Table 4: Experimental settings and hyperparameters for training **LatentNutri** and the supervised baselines.

A.4 Latent Refinement Projection

The projection r_ψ combines a previous-round anchor state $\mathbf{h} \in \mathbb{R}^d$ with the original anchor embedding $\mathbf{e} \in \mathbb{R}^d$ through a learned dimension-wise gate. We first apply a bias-free linear projection and layer normalization to $\mathbf{h}$, then blend the result with $\mathbf{e}$:

$$\begin{aligned} \boldsymbol{\beta} &= \mathrm{sigmoid}(\mathbf{b}_r), \\ r_\psi(\mathbf{h}, \mathbf{e}) &= (\mathbf{1} - \boldsymbol{\beta}) \odot \mathbf{e} + \boldsymbol{\beta} \odot \mathrm{LN}(\mathbf{W}_r \mathbf{h}). \end{aligned} \tag{22}$$

Here $\mathbf{W}_r \in \mathbb{R}^{d \times d}$ is the projection matrix, $\mathbf{b}_r \in \mathbb{R}^d$ is a trainable gate parameter, and $\odot$ denotes element-wise multiplication. The sigmoid is applied element-wise, giving $\boldsymbol{\beta} \in (0, 1)^d$. The projection matrix, layer-normalization parameters, and gate parameter together form ψ and are shared across all anchors and refinement rounds. We initialize $\mathbf{b}_r = \mathbf{0}$, so each gate dimension starts at 0.5.

B Further Analysis

B.1 Physical Plausibility of Predictions

We examine whether predicted nutrition values satisfy basic physical constraints. We report pass rates for energy consistency (EC), mass feasibility (MF), non-negativity (NN), and their conjunction (All). As shown in Table 5, base VLMs can produce numerically plausible text while still violating nutritional relations, especially energy consistency. Autoregressive SFT substantially improves these rates, but **LatentNutri** further raises the all-pass rate to 96.45%, outperforming the Qwen3-VL-4B/8B-SFT baselines and GPT-4.1. Removing the physics regularization lowers the all-pass rate from 96.45% to 93.15%, indicating that SPG contributes not only to numerical accuracy but also to physically coherent predictions. The remaining gap to Gemini-3-Pro suggests that stronger general-purpose VLMs can encode useful dietary priors, while our Anchor-Driven Latent Computation framework makes such constraints explicit during training.

Method	EC (%)	MF (%)	NN (%)	All (%)
Qwen3-VL-4B	68.05	99.41	100.00	67.46
Qwen3-VL-8B	89.15	98.62	100.00	87.97
Qwen3-VL-4B-SFT	95.05	99.85	100.00	94.90
Qwen3-VL-8B-SFT	95.10	99.90	100.00	95.00
GPT-4.1	96.06	99.21	100.00	95.66
Gemini-3-Pro	**97.35**	100.00	100.00	**97.35**
Ours w/o PhysReg	93.65	99.10	100.00	93.15
LatentNutri	96.45	100.00	100.00	96.45

Table 5: Physical plausibility pass rates for model predictions. EC, MF, and NN denote energy consistency, mass feasibility, and non-negativity, while All requires all checks to pass. Higher values are better.

B.2 Effect of Anchor Template Design

We further evaluate whether the ordinary template context around special anchor tokens is useful. Tables 6 and 7 compare the default semantic template with variants that remove template semantics and append either five or one special token at the output tail. These consistent gains across both datasets show that field-aware template semantics, rather than the number of appended anchor tokens alone, are important for effective latent nutrition regression.

B.3 Cross-Family Backbone Results

Table 8 further evaluates the proposed Anchor-Driven Latent Computation framework on different VLM backbone families. The same anchor template, regression heads, and training recipe are applied to Qwen2.5-VL and Qwen3-VL backbones. The method works across both families, and the final accuracy is related to the underlying backbone capability. Qwen2.5-VL-7B improves over Qwen2.5-VL-3B, while Qwen3-VL backbones provide the strongest DiningBench results. This indicates that **LatentNutri** is not tied to a single VLM instance and can use stronger visual-language representations when the backbone provides them.

B.4 Anchor State Visualization Across Backbone Scales

Figure 7 compares t-SNE projections of anchor states across 2B, 4B, and 8B Qwen3-VL backbones. The same anchor mechanism produces target-specific regions at all tested scales, while larger backbones show clearer separation between nutrition targets. This anchor-state trend is consistent with the performance trend.

Method	Calories		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
No template, 1 tail token	181.40	23.89	11.65	31.54	16.25	28.55	12.98	30.56	55.57	28.63
No template, 5 tail tokens	176.10	23.19	11.20	30.32	15.57	27.35	12.36	29.10	53.81	27.49
Default semantic template	171.32	22.56	10.85	29.38	15.13	26.58	11.98	28.20	52.32	26.68

Table 6: Effect of anchor template design on DiningBench. The default semantic template outperforms variants that remove field-specific context around anchor tokens.

Method	Calories		Mass		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
No template, 1 tail token	50.75	19.87	31.10	15.67	4.83	27.65	4.80	24.38	4.10	31.98	19.12	23.91
No template, 5 tail tokens	48.55	19.01	29.70	14.97	4.55	26.06	4.52	22.95	3.85	30.02	18.23	22.60
Default semantic template	46.64	18.26	28.52	14.37	4.33	24.79	4.28	21.73	3.62	28.22	17.48	21.47

Table 7: Effect of anchor template design on Nutrition5K. The default semantic template outperforms variants that remove field-specific context around anchor tokens.

Method	Calories		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
LatentNutri (Qwen2.5-VL-3B)	180.37	23.75	11.23	30.42	16.80	29.51	12.57	29.60	55.24	28.32
LatentNutri (Qwen2.5-VL-7B)	173.32	22.82	10.84	29.35	16.40	28.82	12.19	28.69	53.19	27.42
LatentNutri (Qwen3-VL-4B)	171.32	22.56	10.85	29.38	15.13	26.58	11.98	28.20	52.32	26.68
LatentNutri (Qwen3-VL-8B)	**170.85**	**22.49**	**10.81**	**29.26**	**15.04**	**26.42**	**11.91**	**28.03**	**52.15**	**26.55**

Table 8: DiningBench results for different VLM backbones under the same Anchor-Driven Latent Computation framework. **LatentNutri** transfers across Qwen2.5-VL and Qwen3-VL families.

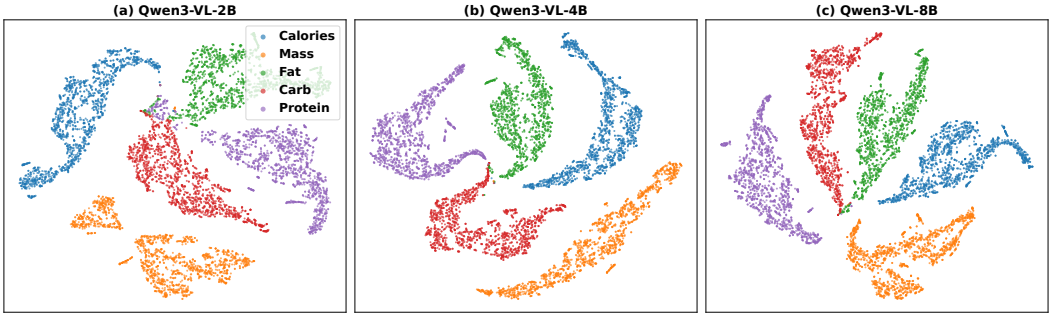


Figure 7: t-SNE comparison of anchor states across Qwen3-VL backbone scales. Each point represents one image-anchor pair and is colored by nutrition target.

B.5 Training Cost and Inference Trade-off

Table 9 compares training costs using the same Qwen3-VL-4B backbone, training data, and four NVIDIA A100 80GB GPUs. **LatentNutri** incurs $4.34\times$ the training GPU-hours of the autoregressive SFT baseline under their respective schedules, including the SFT warm-up for LatentNutri. Beyond the additional warm-up epoch, GPU underutilization during anchor-state extraction and regression-head processing may contribute to this overhead. Further optimization of the training infrastructure, including more efficient hidden-state extraction and operator fusion, could improve GPU utilization and reduce training time. Its efficiency advantage is primarily at inference time: the default single-pass model reduces per-sample latency from 1878.3 ms to 126.4 ms ($14.9\times$ speedup) while improving prediction accuracy. This trades higher one-time training cost for lower inference latency, allowing the additional expense to be amortized over repeated deployment.

Method	**Training schedule**	**Wall-clock (h)**	**GPU-hours**
Qwen3-VL-4B-SFT	3 SFT epochs	2.414	9.657
LatentNutri	1 SFT epoch + 3 latent-training epochs	10.475	41.900

Table 9: Training cost on four NVIDIA A100 80GB GPUs under each method's training schedule. GPU-hours equal wall-clock hours multiplied by four.

C NutriFood-75K Dataset

C.1 Construction Details

NutriFood-75K is designed to provide scalable supervision for learning nutrition-sensitive latent representations, not to replace held-out evaluation benchmarks. The initial image pool combines all available MM-Food-100K images (Dong et al., 2025) with 200 images per category sampled from ISIA Food-500 (Min et al., 2020), giving broad coverage of common food categories and open-domain visual appearances. Because these sources do not provide uniformly reliable per-image nutrition labels, we construct annotations through a VLM-assisted pipeline.

The pipeline contains two quality-control stages. First, Qwen3-VL-8B-Instruct acts as an image inspector and returns strict JSON flags for image clarity, food visibility, whether the food is intact and uneaten, whether the image contains multiple distinct foods, centering, food-area dominance, packaging status, and confidence. Images that fail the visual-quality requirements are removed before nutrition annotation. Second, Gemini-3.1-Pro performs an additional quality review and estimates total calories, total mass, fat, carbohydrates, and protein for the retained images. We discard malformed outputs, low-confidence annotations, and values that violate basic physical constraints, such as negative nutrition quantities, macronutrient mass exceeding total mass, or implausible energy–macronutrient relationships. For MM-Food-100K, we further remove samples whose estimated nutrition values show abnormal disagreement with the source annotations. The exact prompts used for the two-stage construction pipeline are provided in Appendix C.3.

Table 10 reports the number of samples retained after each filtering step for both sources. MM-Food-100K has a higher first-stage rejection rate because it contains more diverse real-world scenes, including multiple dishes, leftovers, and packaged foods. ISIA Food-500 more often contains a clear and dominant food subject. The resulting dataset contains 26,564 MM-Food-100K samples and 48,983 ISIA Food-500 samples, totaling 75,547 training instances.

To assess annotation plausibility, two annotators independently reviewed 100 randomly sampled image–annotation pairs from the final dataset. Drawing on everyday knowledge of food and nutrition, they checked for obvious inconsistencies between the depicted food and the estimated nutrition values. This review revealed no systematic error patterns.

C.2 Dataset Effectiveness

Table 11 evaluates whether the constructed NutriFood-75K provides reliable supervision for nutrition estimation. Compared with the zero-shot Qwen3-VL-4B baseline, SFT on NutriFood-75K reduces average MAE from 46.50 to 30.41 on Nutrition5K and from 101.83 to 55.64 on DiningBench.

Stage	MM-Food-100K			ISIA Food-500		
	Input	Rejected	Remaining	Input	Rejected	Remaining
Initial pool	100,000	—	100,000	100,000	—	100,000
Qwen visual inspection	100,000	58,437	41,563	100,000	8,673	91,327
Gemini visual review	41,563	6,941	34,622	91,327	18,709	72,618
Output-validity and confidence filtering	34,622	2,113	32,509	72,618	12,163	60,455
Physical-plausibility filtering	32,509	2,817	29,692	60,455	11,472	48,983
Source-label consistency filtering	29,692	3,128	26,564	—	—	48,983
Final subset	—	73,436	**26,564**	—	51,017	**48,983**

Table 10: Sample counts throughout NutriFood-75K construction. The final subset contains 75,547 samples in total; rejected counts in the final row are cumulative. Because an image may trigger multiple quality flags, each rejected image is counted once under a primary rejection reason. Source-label consistency filtering applies only to MM-Food-100K.

The gains on both dataset evaluation suggest that the VLM-assisted annotation pipeline produces useful nutrition labels rather than merely adding noisy pseudo labels. When NutriFood-75K is combined with Nutrition5K, the model keeps strong Nutrition5K performance and remains competitive on DiningBench. These results support the quality and practical utility of the constructed dataset.

Training Setting	Nutrition5K MAE	Nutrition5K PMAE (%)	DiningBench MAE	DiningBench PMAE (%)
Qwen3-VL-4B zero-shot	46.50	55.34	101.83	41.52
SFT on Nutrition5K	**17.75**	**21.69**	82.12	47.25
SFT on NutriFood-75K	30.41	35.93	**55.64**	**28.39**
SFT on Nutrition5K + NutriFood-75K	19.13	23.05	56.83	29.14

Table 11: Effect of SFT training data. All SFT rows use the same Qwen3-VL-4B backbone and autoregressive training recipe, varying only the supervision data.

C.3 NutriFood-75K Construction Prompts

The NutriFood-75K construction pipeline employs three prompts across two stages. Stage 1 applies Qwen3-VL-8B-Instruct as a conservative visual quality inspector (Table 12). Stage 2 applies Gemini-3.1-Pro for a secondary quality review (Table 13) and for nutrition value estimation (Table 14).

System Prompt

You are a strict dataset quality inspector for fine-grained food image classification and food image nutrition estimation. You must output ONLY one JSON object that is valid JSON. Do NOT output any extra text. Do NOT use code fences.

The JSON object must contain ALL and ONLY these keys: `image_clear`, `mainly_food`, `food_intact_uneaten`, `multi_food`, `food_centered`, `food_major`, `packaged_unopened`, `confidence`, `reasons`. `confidence` must be between 0 and 1. `reasons` must be in English, 1 to 3 short sentences.

Be conservative: if uncertain, set the corresponding "good-when-true" booleans to false. `image_clear` / `mainly_food` / `food_intact_uneaten` / `food_centered` / `food_major` are "true = good". `multi_food` and `packaged_unopened` are "true = filter-out signals". This is for filtering: prefer the more pessimistic value when quality is questionable.

User Prompt

You are filtering images for a high-quality food classification & nutrition dataset. Goal: keep ONLY images that show a single, clearly visible, fully framed, uneaten main food/dish.

Decide each boolean independently and strictly:
(A) `image_clear` (true = good): True ONLY if the image is clear enough to recognize the main subject and to estimate nutrition. False if blurry, pixelated, very low resolution, or severely under/overexposed.
(B) `mainly_food` (true = good): True ONLY if the main content is actually food/meal/drink. False for people, pets, scenery, menus, screenshots, receipts, logos, etc.
(C) `food_intact_uneaten` (true = good): True ONLY if the main food is intact and NOT eaten. False if there are obvious bites, missing portions, or leftovers.
(D) `multi_food` (true = filter-out): True if multiple clearly different foods/dishes coexist. Multiple portions of the SAME food is NOT `multi_food`.
(E) `food_centered` (true = good): True ONLY if the main food is roughly centered and fully shown. False if heavily off-center, at an edge, or truncated.
(F) `food_major` (true = good): True ONLY if the main food occupies most of the image area. False if too small or the background dominates.
(G) `packaged_unopened` (true = filter-out): True if mainly unopened packaging is shown and the actual food is NOT visible. False if opened and food is clearly visible.

Provide `confidence` (a number in $[0, 1]$) and `reasons` (1–3 short English sentences on any non-default decisions). Output STRICT JSON only. No extra keys, no extra text, no code fences.

Table 12: Stage-1 image quality filtering prompt for **Qwen3-VL-8B-Instruct**. The same prompt checks seven binary quality dimensions for candidate images from both MM-Food-100K and ISIA Food-500.

System Prompt

You are a strict dataset quality inspector for food image nutrition estimation. You must output ONLY one JSON object that is valid JSON. Do NOT output any extra text. Do NOT use code fences.

The JSON object must contain ALL and ONLY these keys: `image_clear`, `mainly_food`, `food_intact_uneaten`, `multi_food`, `food_centered`, `food_major`, `packaged_unopened`, `confidence`, `reasons`. `confidence` must be between 0 and 1. `reasons` must be in English, 1 short sentence. Be conservative: prefer false over true when quality is questionable.

User Prompt

You are filtering images for a food nutrition estimation dataset (ISIA_Food500 / fine-grained class). Apply STRICT quality rules. We want: a single main food/meal, clearly visible, centered, fully shown, uneaten, and informative.

Decide the following booleans:
(A) `image_clear`: True ONLY if the image is clear enough for nutrition estimation.
(B) `mainly_food`: True ONLY if the main content is actually food/meal.
(C) `food_intact_uneaten`: True ONLY if the main food appears intact and NOT eaten.
(D) `multi_food`: True if there are multiple clearly different foods/dishes/plates in the same image.
(E) `food_centered`: True ONLY if the main food is roughly centered and not heavily cropped.
(F) `food_major`: True ONLY if the main food occupies most of the image attention/area.
(G) `packaged_unopened`: True if it is mainly unopened packaged food / only packaging.

Output STRICT JSON only. No extra keys, no extra text. Keep reasons very short.

Table 13: Stage-2 image quality review prompt for **Gemini-3.1-Pro**. The prompt applies the same seven binary quality flags to retained candidates.

System Prompt

You are a nutrition estimation assistant for a single food dish image. You must output ONLY one JSON object, valid JSON, with numeric values. Do NOT output any other words, headers, prefixes, or code fences.

User Prompt

Look at this food dish image. Reply with ONLY ONE JSON object containing exactly these keys and numeric values (no units, no strings):

`total_calories`, `total_mass`, `total_fat`, `total_carb`, `total_protein`.

Table 14: Stage-2 nutrition annotation prompt for **Gemini-3.1-Pro**. The model estimates five nutrition values from the food image and returns a numeric JSON object.

D CASE STUDIES

Figures 8–11 provide four representative qualitative examples. Each figure shows the food image, ground-truth nutrition values, predictions from Google-Nutrition-rgb, Qwen3-VL-4B, Qwen3-VL-4B-SFT, and **LatentNutri**. The examples cover seafood, congee, rice bowl, and sashimi dishes with different visual appearances and nutrient compositions. Across these cases, the autoregressive VLM baselines often produce large deviations on one or more macronutrients, even when the food category is visually recognizable. In contrast, **LatentNutri** gives more balanced predictions across calories, protein, carbohydrates, and fat, resulting in lower average PMAE. These cases illustrate that anchor states can support more stable latent nutrition computation than decoding values as text.

Grilled Seafood Platter



Method	Calories (kcal)	Protein (g)	Carb (g)	Fat (g)	Avg. PMAE (%)
Ground Truth (reference)	720	72.0	8.0	44.0	—
Google-Nutrition-rgb	656	48.9	34.2	31.7	99.1
Qwen3-VL-4B	800	60.0	30.0	70.0	90.5
Qwen3-VL-4B-SFT	1050	115.0	15.0	55.0	54.5
LatentNutri	**816**	**81.5**	**9.9**	**43.5**	**12.8**

Figure 8: Case study on a grilled seafood platter, comparing ground-truth nutrition values with predictions from representative models.

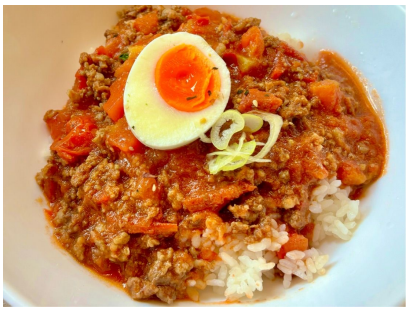
Shrimp Congee



Method	Calories (kcal)	Protein (g)	Carb (g)	Fat (g)	Avg. PMAE (%)
Ground Truth (reference)	272	16.0	42.0	4.0	—
Google-Nutrition-rgb	224	17.0	24.0	3.8	17.8
Qwen3-VL-4B	350	25.0	55.0	15.0	97.7
Qwen3-VL-4B-SFT	320	20.0	45.0	6.0	24.9
LatentNutri	**308**	**19.5**	**44.8**	**4.7**	**14.9**

Figure 9: Case study on shrimp congee, comparing ground-truth nutrition values with predictions from representative models.

Tomato Beef Donburi



Method	Calories (kcal)	Protein (g)	Carb (g)	Fat (g)	Avg. PMAE (%)
Ground Truth (reference)	615	24.0	72.0	26.0	—
Google-Nutrition-rgb	689	26.7	85.3	24.9	11.4
Qwen3-VL-4B	450	30.0	70.0	25.0	14.6
Qwen3-VL-4B-SFT	650	28.0	70.0	28.0	8.2
LatentNutri	**648**	**28.0**	**70.0**	**27.5**	**7.6**

Figure 10: Case study on tomato beef donburi, comparing ground-truth nutrition values with predictions from representative models.

Assorted Fish Sashimi Platter



Method	Calories (kcal)	Protein (g)	Carb (g)	Fat (g)	Avg. PMAE (%)
Ground Truth (reference)	390	53.0	4.0	17.0	—
Google-Nutrition-rgb	690	68.7	10.1	38.8	96.8
Qwen3-VL-4B	500	50.0	30.0	35.0	197.4
Qwen3-VL-4B-SFT	520	40.0	45.0	18.0	272.2
LatentNutri	**416**	**53.3**	**8.5**	**11.3**	**38.4**

Figure 11: Case study on an assorted fish sashimi platter, comparing ground-truth nutrition values with predictions from representative models.

E Extended Experimental Results

E.1 Full DiningBench Results

Table 15 reports full per-target results for all 18 methods on DiningBench, extending the representative comparison in Table 1.

E.2 Full Nutrition5K Results

Table 16 reports full per-target results for all methods on Nutrition5K.

E.3 Backbone Scale Per-Target Results

Tables 17 and 18 provide per-target breakdowns for the backbone scale comparison discussed in the *Effect of VLM backbone scale* analysis in Section 4.5.

E.4 Latent Refinement Per-Target Results

Tables 19 and 20 provide per-target breakdowns for the refinement round analysis discussed in the *Effect of test-time latent refinement* analysis in Section 4.5.

F Prompts

We provide the prompt used for zero-shot VLM evaluation.

Method	Calories		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
CNN/ViT baselines										
ResNet-101	245.6	32.3	16.04	43.4	27.94	49.1	16.78	39.5	76.58	41.10
ConvNeXt-B	226.4	29.8	14.31	38.8	25.76	45.3	15.57	36.6	70.51	37.60
Swin-B	232.1	30.6	14.68	39.8	25.34	44.5	16.82	39.6	72.22	38.60
Google-Nutrition-rgb	255.9	33.7	15.39	41.7	29.62	52.0	17.51	41.2	79.62	42.20
Zero-shot VLMs										
Gemini-2.5-Pro	229.8	30.3	13.50	36.6	16.68	29.3	18.14	42.7	69.52	34.70
Gemini-3-Pro	<u>175.2</u>	<u>23.1</u>	**10.42**	**28.2**	**13.95**	**24.5**	**10.83**	**25.5**	<u>52.60</u>	**25.30**
GPT-4.1	312.4	41.1	14.24	38.6	17.93	31.5	23.48	55.3	92.00	41.61
Qwen3-VL-4B	353.6	46.6	14.27	38.6	20.29	35.7	19.21	45.2	101.83	41.52
Qwen3-VL-8B	264.0	34.8	14.00	37.9	19.41	34.1	19.20	45.2	79.16	37.99
Qwen2.5-VL-3B	352.7	46.4	15.32	41.5	30.24	53.1	25.97	61.1	106.05	50.55
Qwen2.5-VL-7B	331.3	43.6	15.66	42.4	28.76	50.5	25.37	59.7	100.27	49.07
InternVL3.5-30B-A3B	332.9	43.8	16.49	44.7	25.24	44.3	22.53	53.0	99.28	46.46
SFT VLMs										
Qwen3-VL-4B-SFT	185.7	24.4	11.67	31.6	16.87	29.6	13.12	30.9	56.83	29.14
Qwen3-VL-8B-SFT	183.2	24.1	11.23	30.4	16.02	28.1	12.96	30.5	55.85	28.29
Qwen2.5-VL-3B-SFT	203.5	26.8	12.62	34.2	18.71	32.9	14.72	34.6	62.38	32.12
Qwen2.5-VL-7B-SFT	186.0	24.5	11.97	32.4	17.46	30.7	13.27	31.2	57.19	29.70
FoodLMM	205.0	27.0	11.78	31.9	21.11	37.1	14.30	33.7	63.05	32.41
Ours										
LatentNutri	**171.3**	**22.6**	<u>10.85</u>	<u>29.4</u>	<u>15.13</u>	<u>26.6</u>	<u>11.98</u>	<u>28.2</u>	**52.32**	<u>26.68</u>

Table 15: Full results on DiningBench. MAE and PMAE are reported for each nutrition target and macro-averaged across targets. Best and second-best results are shown in **bold** and <u>underlined</u>.

Method	Calories		Mass		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
CNN/ViT baselines												
ResNet-101	59.7	23.4	38.6	19.5	6.12	35.0	7.12	36.2	4.55	35.5	23.22	29.90
ConvNeXt-B	65.4	25.6	40.2	20.3	6.47	37.0	7.29	37.0	5.17	40.3	24.91	32.00
Swin-B	63.2	24.7	40.8	20.6	6.65	38.1	8.43	42.9	5.03	39.2	24.83	33.10
Google-Nutrition-rgb	70.6	26.1	40.4	18.8	5.50	29.5	6.10	31.9	5.00	34.2	25.52	28.10
Zero-shot VLMs												
Gemini-2.5-Pro	88.0	34.5	58.7	29.6	6.40	36.6	7.27	36.9	6.88	53.6	33.45	38.24
Gemini-3-Pro	71.8	28.1	49.9	25.1	5.58	31.9	6.42	32.6	5.56	43.3	27.85	32.20
GPT-4.1	80.0	31.3	65.5	33.0	7.53	43.1	7.20	36.6	6.19	48.2	33.28	38.45
Qwen3-VL-4B	123.9	48.5	75.8	38.2	8.19	46.9	17.89	90.9	6.70	52.2	46.50	55.34
Qwen3-VL-8B	109.6	42.9	77.5	39.0	8.73	50.0	15.38	78.1	7.15	55.8	43.67	53.15
Qwen2.5-VL-3B	112.9	44.2	81.7	41.2	11.27	64.5	20.33	103.3	11.69	91.1	47.58	68.84
Qwen2.5-VL-7B	115.2	45.1	96.0	48.4	9.71	55.6	16.09	81.7	7.05	55.0	48.81	57.15
InternVL3.5-30B-A3B	93.2	36.5	75.3	37.9	8.53	48.8	9.68	49.2	7.11	55.4	38.75	45.56
SFT VLMs												
Qwen3-VL-4B-SFT	51.1	20.0	31.5	15.8	4.52	25.8	4.83	24.6	3.72	29.0	19.13	23.05
Qwen3-VL-8B-SFT	49.6	19.4	30.8	15.5	4.16	23.8	4.58	23.3	3.81	29.7	18.58	22.33
Qwen2.5-VL-3B-SFT	56.3	22.0	35.6	17.9	5.18	29.6	5.43	27.6	4.06	31.6	21.32	25.77
Qwen2.5-VL-7B-SFT	52.1	20.4	33.9	17.1	4.98	28.5	5.04	25.6	3.88	30.3	19.97	24.36
FoodLMM	67.3	26.6	39.7	20.7	4.10	25.8	5.90	31.1	5.40	39.7	24.48	28.80
Ours												
LatentNutri	46.6	18.3	28.5	14.4	4.33	24.8	4.28	21.7	3.62	28.2	17.48	21.47

Table 16: Full target-level results on Nutrition5K. MAE and PMAE are reported for calories, mass, protein, carbohydrate, fat, and their macro average.

Method	Calories		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
LatentNutri (Qwen3-VL-2B)	177.83	23.41	10.91	29.55	16.00	28.11	12.17	28.64	54.23	27.43
LatentNutri (Qwen3-VL-4B)	171.32	22.56	10.85	29.38	15.13	26.58	11.98	28.20	52.32	26.68
LatentNutri (Qwen3-VL-8B)	170.85	22.49	10.81	29.26	15.04	26.42	11.91	28.03	52.15	26.55

Table 17: Per-target DiningBench results across Qwen3-VL backbone scales. MAE and PMAE are reported for 2B, 4B, and 8B variants.

Method	Calories		Mass		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
LatentNutri (Qwen3-VL-2B)	46.12	18.05	30.72	15.48	4.05	23.14	4.59	23.3	3.94	30.72	17.88	22.14
LatentNutri (Qwen3-VL-4B)	46.64	18.26	28.52	14.37	4.33	24.79	4.28	21.73	3.62	28.22	17.48	21.47
LatentNutri (Qwen3-VL-8B)	46.42	18.17	28.34	14.28	4.29	24.55	4.22	21.45	3.60	28.04	17.37	21.30

Table 18: Per-target Nutrition5K results across Qwen3-VL backbone scales. MAE and PMAE are reported for 2B, 4B, and 8B variants.

Round	Calories		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
1 round	171.32	22.56	10.85	29.38	15.13	26.58	11.98	28.20	52.32	26.68
2 rounds	169.95	22.38	10.66	28.88	14.86	26.10	11.74	27.65	51.80	26.25
3 rounds	169.35	22.30	10.58	28.67	14.70	25.82	11.62	27.37	51.56	26.04
4 rounds	169.64	22.34	10.61	28.75	14.74	25.89	11.66	27.46	51.66	26.11
5 rounds	169.88	22.37	10.65	28.86	14.82	26.03	11.72	27.60	51.77	26.22

Table 19: Per-target DiningBench results across latent refinement rounds. The table provides the target-level breakdown for the average-PMAE trends in Figure 6.

Round	Calories		Mass		Protein		Carb.		Fat		Avg.	
	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)	MAE	PMAE(%)
1 round	46.64	18.26	28.52	14.37	4.33	24.79	4.28	21.73	3.62	28.22	17.48	21.47
2 rounds	45.50	17.80	27.03	13.62	4.13	23.65	4.17	21.18	3.60	28.03	16.89	20.86
3 rounds	45.10	17.65	26.83	13.52	4.08	23.36	4.12	20.91	3.56	27.78	16.74	20.64
4 rounds	45.00	17.61	26.75	13.48	4.07	23.29	4.10	20.84	3.56	27.70	16.70	20.58
5 rounds	44.90	17.57	26.67	13.44	4.05	23.19	4.08	20.75	3.55	27.60	16.65	20.51

Table 20: Per-target Nutrition5K results across latent refinement rounds. The table provides the target-level breakdown for the average-PMAE trends in Figure 6.

F.1 ZERO-SHOT VLM EVALUATION PROMPT

Table 21 shows the prompt used for all zero-shot VLM baselines.

User Prompt
Look at this food dish image. Predict the following nutritional values as numbers only. Reply with a single JSON object containing exactly these keys and numeric values: `total_calories`, `total_mass`, `total_fat`, `total_carb`, `total_protein`. No other text, only the JSON.

Table 21: Zero-shot VLM evaluation prompt used for all zero-shot baselines.

G LIMITATIONS & POTENTIAL RISKS

LatentNutri has several limitations and potential risks. First, nutrition estimation from a single food image requires inferring factors that are only partially observable, such as hidden ingredients, cooking oil, sauce composition, exact portion weight, or preparation details. As a result, despite improved benchmark accuracy, predictions may still be affected by visual ambiguity and noise in nutrition annotations. Second, the current framework is instantiated with a fixed anchor template and target-specific regression heads for calories, mass, fat, carbohydrates, and protein. Adapting it to additional nutrients, allergens, ingredient quantities, or user-defined dietary attributes would require corresponding anchors, supervision, and validation. The primary risk is over-reliance on automatic estimates in health-sensitive contexts. Such predictions should be viewed as decision-support signals rather than substitutes for professional dietary assessment, particularly for users with medical conditions, allergies, eating disorders, or strict nutritional requirements. Without appropriate uncertainty communication or human review, inaccurate estimates may lead to inappropriate dietary decisions.