EgoCal: Combining Egocentric Data with Vision-Language Fusion for Real-time Dietary Tracking

Supplementary Material

7. Evaluation Metrics Formulation

To rigorously define our evaluation metrics, let N be the total number of eating sequences in the dataset. For a given sequence $i \in \{1, \ldots, N\}$, let T_i represent the total number of valid frames (initial state plus subsequent bites). We define $W_{i,t}$ as the ground-truth weight and $\hat{W}_{i,t}$ as the predicted weight for sequence i at time step $t \in \{0, \ldots, T_i - 1\}$, where $t = 0$ corresponds to the initial frame.

To evaluate caloric estimation, let D_i denote the caloric density (kCal/g) of the specific food class present in sequence i.

7.1. Weight Estimation Metrics

The Mean Absolute Error (MAE) and Percentage Mean Absolute Error (PMAE) are calculated across three distinct temporal granularities to capture different aspects of the tracking performance. PMAE normalizes the absolute error by the ground-truth mean of that specific evaluation subset.

Overall Weight Error. Measures the frame-by-frame accuracy across the entire dataset:

$$\text{MAE}_{\text{Overall}} = \frac{1}{\sum_{i=1}^{N} T_i} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} \left| W_{i,t} - \hat{W}_{i,t} \right| \quad (5)$$

$$\text{PMAE}_{\text{Overall}} = \frac{\text{MAE}_{\text{Overall}}}{\frac{1}{\sum_{i=1}^{N} T_i} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} W_{i,t}} \times 100 \quad (6)$$

Initial Frame Weight Error. Isolates the absolute prediction accuracy at the start of the meal ($t = 0$):

$$\text{MAE}_{\text{Initial}} = \frac{1}{N} \sum_{i=1}^{N} \left| W_{i,0} - \hat{W}_{i,0} \right| \quad (7)$$

$$\text{PMAE}_{\text{Initial}} = \frac{\text{MAE}_{\text{Initial}}}{\frac{1}{N} \sum_{i=1}^{N} W_{i,0}} \times 100 \quad (8)$$

Consumed Weight Error. Evaluates the total mass removed from the plate over the course of the entire sequence:

$$\text{MAE}_{\text{Consumed}} = \frac{1}{N} \sum_{i=1}^{N} \left| (W_{i,0} - W_{i,T_i-1}) - (\hat{W}_{i,0} - \hat{W}_{i,T_i-1}) \right| \quad (9)$$

$$\text{PMAE}_{\text{Consumed}} = \frac{\text{MAE}_{\text{Consumed}}}{\frac{1}{N} \sum_{i=1}^{N} (W_{i,0} - W_{i,T_i-1})} \times 100 \quad (10)$$

7.2. Caloric Estimation Metrics

To measure the nutritional impact of the tracking errors, the weight errors are scaled by the energy density D_i of the respective food classes. Because caloric density varies heavily by category, this compounds weight errors asymmetrically.

Overall Caloric Error. Measures the total energy estimation error across all frames:

$$\text{MAE}_{\text{kCal}} = \frac{1}{\sum_{i=1}^{N} T_i} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} \left| W_{i,t} - \hat{W}_{i,t} \right| \times D_i \quad (11)$$

$$\text{PMAE}_{\text{kCal}} = \frac{\text{MAE}_{\text{kCal}}}{\frac{1}{\sum_{i=1}^{N} T_i} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} W_{i,t} \times D_i} \times 100 \quad (12)$$

8. Additional Details for VLM Baselines

In this section, we provide the exact prompts used for our VLM baseline comparisons and relevant inference details. In our experiments, each frame was passed to the models independently where the model was asked to estimate the remaining food weight in grams. As mentioned, we evaluate two prompting controls: (1) providing or not providing the semantic prior in the food name, and (2) direct numeric prediction versus a reasoning-style prompt asking the model to consider visual cues such as perceived volume, surface area, and thickness of the food prior to providing its estimate. In all of our cases, the model was asked to constrain its output to a single numeric estimate. We provide each prompt in Listings 1 - 4.

Listing 1. Direct Prediction Prompt (With Prior)

```
"Estimate the remaining weight of the food
    in this image in grams.

The food item in this image is [FOOD_NAME].

Use your knowledge of this food's typical
    density and portion characteristics to
    improve your estimate.

Respond with ONLY a single numeric value (
    the estimated weight in grams).

Do not include units, explanations, or any
    other text. Example valid responses:
    245.3 or 180"
```

Listing 2. Direct Prediction Prompt (Without Prior)

```
"Estimate the remaining weight of the food
    in this image in grams.

Respond with ONLY a single numeric value (
    the estimated weight in grams).

Do not include units, explanations, or any
    other text. Example valid responses:
    245.3 or 180"
```

Listing 3. Reasoning Prediction Prompt (Without Prior)

```
"You are an expert dietician.

Given the following image of a food item on
     a plate, estimate the remaining weight
    of the food in grams.

Analyze the visual volume, surface area,
    and apparent thickness of the food to
    produce your best estimate.

Respond with ONLY a single numeric value (
    the estimated weight in grams).

Do not include units, explanations, or any
    other text. Example valid responses:
    245.3 or 180"
```

Listing 4. Reasoning Prediction Prompt (With Prior)

```
"You are an expert dietician.

Given the following image of a food item on
     a plate, estimate the remaining weight
    of the food in grams.

The food item in this image is [FOOD\_NAME
    ].

Use your knowledge of this food's typical
    density and portion characteristics to
    improve your estimate.

Analyze the visual volume, surface area,
    and apparent thickness of the food to
    produce your best estimate.

Respond with ONLY a single numeric value (
    the estimated weight in grams). Do not
    include units, explanations, or any
    other text. Example valid responses:
    245.3 or 180"
```

For local model inference, we used a single NVIDIA A40, the huggingface `transformers` API, and deterministic output decoding (`do_sample=False`). If the model response could not parse a single output numeric value, we issued a follow-up prompt requesting strictly a numeric answer:

Listing 5. Follow-Up Prompt for Single Numeric Value Output

```
The previous response could not be parsed
    as a single number. Please respond with
    ONLY a single numeric value representing
     the estimated weight in grams. Nothing
    else. Example: 245.3
```