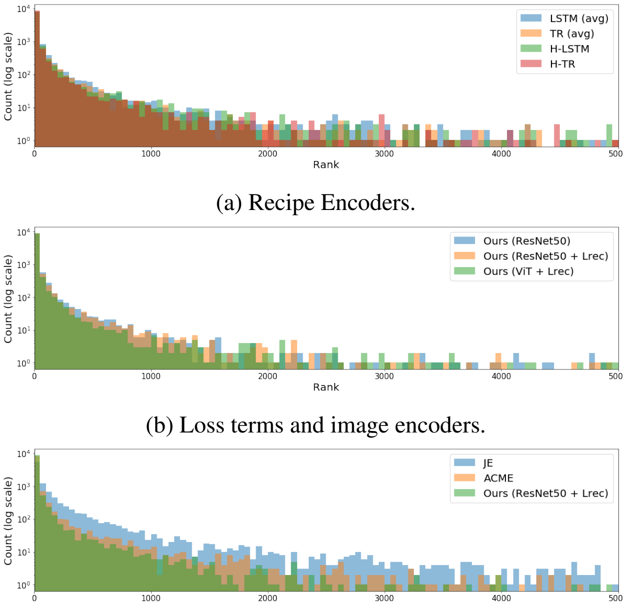
Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning (Supplementary Material)

Amaia Salvador Erhan Gundogdu Loris Bazzani Michael Donoser
Amazon
{asalvada, eggundog, bazzanil, donoserm}@amazon.com

This document provides additional quantitative and qualitative results. In Section 1, we include rank histograms for different image-to-recipe methods, which provide a complementary quantitative comparison between methods. Section 2 extends the missing data experiment from the main paper, with an additional baseline to compare with our approach. Finally, Section 3 presents additional qualitative results compared to those achieved using the pre-trained embeddings from ACME [2].

1. Rank Distribution

In this section, we provide further quantitative results by visualizing the distribution of the obtained ranks for different methods. All histograms are computed using a fixed 10k-sized sample from the test set on the image-to-recipe task. We display histograms of 100 bins and truncate them at rank 5000. Figure 1a shows the histograms of models using different LSTM and Transformer based recipe encoders compared in the paper. Figure 1b compares the histograms for progressive improvements of our model $(\mathcal{L}_{pair})$, $(\mathcal{L}_{pair}+\mathcal{L}_{rec})^{\diamond}$, and $(\mathcal{L}_{pair}+\mathcal{L}_{rec}\text{ (ViT)})^{\diamond}$. Finally, in Figure 1c we display histograms comparing our method with two existing methods in the literature of image-to-recipe retrieval [1, 2] for which code was made available by authors. Figures show the improvement of our model with respect to baselines and previous works, with steeper histograms for our model variants (higher values for low rank bins, lower values for high rank bins).



(a) Recipe Encoders.

(b) Loss terms and image encoders.

(c) Comparison with existing works (JE [1] and ACME [2]).

Figure 1: **Rank histograms** for models trained with different recipe encoders (a), different losses and image encoders (b), and comparison to existing methods in the literature (c).

2. Hallucination Experiments

In this section, we extend Table 4 from the main manuscript by including an additional baseline tested on the missing data scenario. In total, we compare 3 methods, which are described below:

(1) *Hallucinated* e_a (e.g. *Hallucinated* e_{ttl}). Our proposed model trained with $\mathcal{L}_{pair} + \mathcal{L}_{rec}$ using projection functions $g(\cdot)$ to compute $\mathcal{L}_{rec}$ between embeddings from different recipe components. This model is trained using all recipe components, and at test time, embeddings from missing components are hallucinated from the existing ones (e.g. e_{ttl} is replaced with $(g_{ing\rightarrow ttl}(e_{ing}) + g_{ins\rightarrow ttl}(e_{ins}))/2)$.

(2) $\phi_{mrg}(avg(\cdot))$: A model trained with $\mathcal{L}_{pair}$ using average operation to merge the embeddings of individual recipe components (as opposed to embedding concatenation, which is used by default in all our models). By using averaging as the operator to merge embeddings from different recipe components, this model can naturally deal with missing data at test time (i.e. the average can be computed using the available embeddings). We note that this model gives slightly worse performance with respect to concatenation in the original image-to-recipe task (decrease of 0.3 R1 points with respect to concatenation).

	medR	**R1**	**R5**	**R10**
No title ○	6.0	22.7	48.4	60.4
Ignore e_{ttl} ($\phi_{mrg}(avg(\cdot))$)	6.7	21.6	46.6	58.5
Hallucinated e_{ttl}	**5.0**	**24.2**	**51.2**	**63.1**
No ingredients ○	10.2	16.0	38.3	50.2
Ignore e_{ing} ($\phi_{mrg}(avg(\cdot))$)	21.9	9.8	26.5	37.1
Hallucinated e_{ing}	**10.1**	**16.6**	**39.1**	**50.8**
No instructions ○	6.0	22.3	48.0	59.8
Ignore e_{ins} ($\phi_{mrg}(avg(\cdot))$)	8.0	19.3	43.3	55.1
Hallucinated e_{ins}	6.0	**23.1**	**49.4**	**61.1**
Title only ○	**35.5**	6.0	18.9	28.4
Ignore e_{ing}, e_{ins} ($\phi_{mrg}(avg(\cdot))$)	98.5	3.0	10.8	17.2
Hallucinated e_{ing}, e_{ins}	35.8	**6.6**	**20.0**	**29.3**
Ingredients only ○	8.3	19.2	42.5	53.9
Ignore e_{ttl}, e_{ins} ($\phi_{mrg}(avg(\cdot))$)	17.2	13.1	32.1	42.2
Hallucinated e_{ttl}, e_{ins}	**8.0**	**19.4**	**43.5**	**55.3**
Instructions only ○	15.0	13.1	32.6	43.8
Ignore e_{ttl}, e_{ing} ($\phi_{mrg}(avg(\cdot))$)	50.7	5.6	17.4	25.4
Hallucinated e_{ttl}, e_{ing}	**13.9**	**14.0**	**34.1**	**45.4**

Table 1: **Dealing with missing data.** Image-to-recipe retrieval results reported on the test set of Recipe1M. Results reported on rankings of size $10k$. ○ indicates missing component is not used in training nor testing (as opposed to the component being missing only at test time).

(3) *No X/X only* (e.g. *No title*, *Ingredients only*). Models trained and tested by ignoring or only using particular recipe components (the corresponding encoder is removed both during training and testing).

Table 1 includes the retrieval results for all the above models in the missing data scenario. These results demonstrate that our model ($\mathcal{L}_{pair} + \mathcal{L}_{rec}$) not only achieves state of the art performance in the standard image-to-recipe retrieval task, but is also able outperform the baselines in the missing data scenarios. These results indicate that our model, when trained using the additional $\mathcal{L}_{rec}$ is robust when tested on missing data scenarios, suggesting its usefulness for applications in which data is incomplete (e.g. indexing and searching for recipes for which only titles are available).

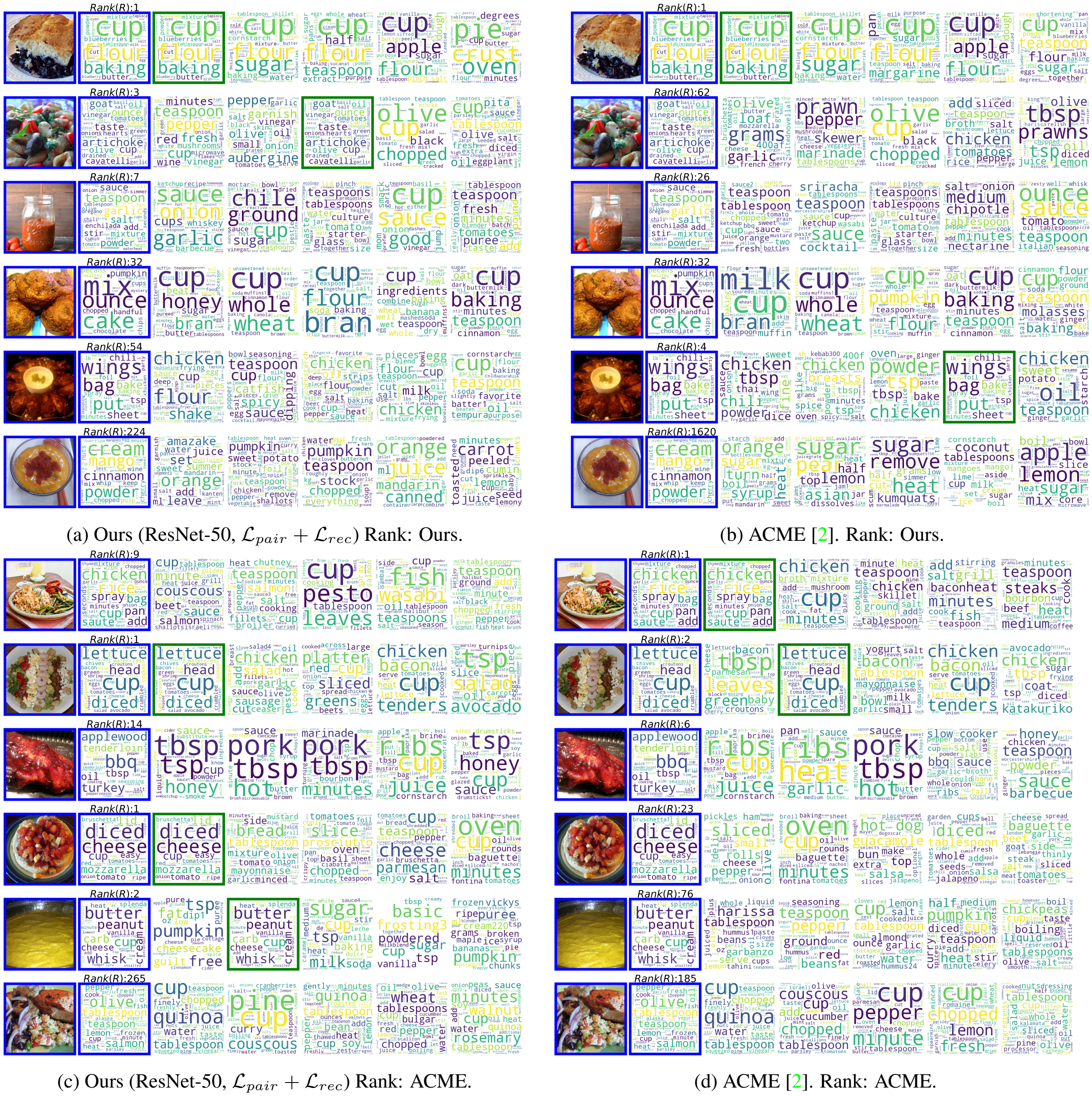
3. Qualitative Results

We include additional qualitative results in the image-to-recipe retrieval task, comparing our method with ACME [2], the best performing image-to-recipe retrieval method for which code was made available. For a fair qualitative comparison, we pick samples to display according to the rank in which the true recipe was found, and randomly select a sample with a rank within the range $R[i-1] < r <= R[i]$, with $R = (1, 5, 10, 50, 100, 500)$. For consistency, we apply this procedure in a bidirectional manner (i.e. we select samples based on our method's ranks and display the results for both methods, and vice-versa). In Figures 2a and 2b, we compare our method with ACME [2] for samples selected following the rank obtained using our model, while results in Figures 2c and 2d were obtained by sampling following the ranks of ACME [2]. Out of the 12 samples selected following the procedure described above, our method achieves superior performance in 6/12 (c.f. 4/12 ACME, and 2/12 tie).

References

[1] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. Learning cross-modal embeddings for cooking recipes and food images. In *CVPR*, 2017. 1

[2] Hao Wang, Doyen Sahoo, Chenghao Liu, Ee-peng Lim, and Steven CH Hoi. Learning cross-modal embeddings with adversarial networks for cooking recipes and food images. In *CVPR*, 2019. 1, 2, 3



(a) Ours (ResNet-50, $\mathcal{L}_{pair} + \mathcal{L}_{rec}$) Rank: Ours.

(b) ACME [2]. Rank: Ours.

(c) Ours (ResNet-50, $\mathcal{L}_{pair} + \mathcal{L}_{rec}$) Rank: ACME.

(d) ACME [2]. Rank: ACME.

Figure 2: **Qualitative comparison.** Top 5 retrieved recipes for different queries obtained with our method (a,c) and ACME (b, d). Samples are chosen randomly within different rank ranges (sorted in descending order), following the rank distribution of our method (a, b), and ACME's (c, d). Each row includes the query (image and recipe, highlighted in blue), followed by the top $K = 5$ retrieved recipes. The correct retrieved element is highlighted in green. For all rankings, the rank of the true recipe $Rank(R)$ is provided for reference.