TIPSv2:
Advancing Vision-Language Pretraining with Enhanced Patch-Text Alignment

Supplementary Material

Bingyi Cao* Koert Chen* Kevis-Kokitsi Maninis Kaifeng Chen† Arjun Karpur†
Ye Xia Sahil Dua Tanmaya Dabral Guangxing Han Bohyung Han† Joshua Ainslie
Alex Bewley Mithun Jacob René Wagner Washington Ramos† Krzysztof Choromanski
Mojtaba Seyedhosseini Howard Zhou André Araujo
Google DeepMind

Table 1. **Applying iBOT++ to CLIP**, on a ViT-L backbone, using the head-only EMA setup. iBOT++ significantly enhances CLIP performance across several tasks, beyond what can be obtained with iBOT.

Model ViT-L	Seg. ADE20k	Depth ↓ NYUv2	ImageNet KNN	IT Ret. COCO	TI Ret. COCO	0-shot Seg. ADE150
CLIP	35.7	0.571	73.5	52.2	33.2	1.2
CLIP+iBOT	39.8	0.467	76.6	55.4	37.0	1.6
CLIP+iBOT++	**42.5**	**0.437**	**78.2**	**56.6**	**39.3**	**7.8**

Table 2. **Applying iBOT++ to CLIP with a dual CLS setup**, on a ViT-g backbone. iBOT++ significantly enhances performance across several tasks, beyond what can be obtained with iBOT.

Model ViT-g	Seg. ADE20k	Depth ↓ NYUv2	ImageNet KNN	IT Ret. COCO	TI Ret. COCO	0-shot Seg. ADE150
CLIP 2 CLS	41.6	0.475	79.1	72.9	56.1	4.6
CLIP 2 CLS + iBOT	41.1	0.442	81.6	73.0	48.3	5.1
CLIP 2 CLS + iBOT++	**47.7**	**0.366**	**82.7**	**73.6**	**58.9**	**18.2**

A. Appendix

A.1. Applying iBOT++ to CLIP

In the main paper, we showed the value of our novel iBOT++ loss in the context of the TIPS training recipe. In this section, we go further to demonstrate the generalizability of iBOT++ to other vision-language architectures, specifically on top of the widely adopted CLIP model. First, both iBOT and iBOT++ are applied on top of vanilla CLIP, trained with the same dataset used throughout the paper, using a ViT-L backbone, also using the head-only EMA setup to reduce memory requirements. Our results in Tab. 1 demonstrate that integrating iBOT++ yields consistent performance gains across all metrics, beyond that of adding iBOT alone. Then, as an additional experiment to validate iBOT++, we add it on top of CLIP using two CLS tokens (as in the TIPS setup, but without any of the additional losses used in TIPS), with a ViT-g backbone and the same training dataset. Once again, we see significant and consistent performance gains using iBOT++, compared to the CLIP 2 CLS baseline and its modified version using iBOT. Notably, iBOT++ yields significant improvements in zero-shot segmentation, highlighting its importance for aligning image patches to language. These results corroborate the usefulness of our proposed iBOT++ recipe for vision-language pretraining.

Table 3. **Masking ablation for iBOT++ pretraining.** We conduct an ablation study on TIPS [3] ViT-L models to determine the optimal masking ratio for iBOT++ pretraining. The results show that 75% masking ratio is critical for achieving strong performance across all evaluations, particularly enhancing patch-text alignment. The model achieving the best overall performance is highlighted.

Masking Ratio	Segmentation ↑ PASCAL	Depth ↓ NYUv2	I→T Ret. ↑ Flickr	Zero-shot Seg. ↑ ADE150
0.0	78.8	0.513	90.0	1.0
0.5	82.5	0.438	92.6	2.4
0.75	**82.6**	**0.418**	**93.8**	**13.6**

A.2. iBOT++ Ablation Study on Masking Ratios

As detailed in Section 3.2, distillation significantly enhances patch-text alignment, even when the teacher model exhibits poor dense alignment. In Table 2 (from the main paper), we identified two key distinctions between the distillation and pretraining stages: A) also applying loss to visible tokens (compare rows (1-2)) and B) lowering masking

*Authors contributed equally.

†Kaifeng Chen is now with xAI, Arjun Karpur with Epsilon Health, Bohyung Han with Seoul National University and Washington Ramos with Google.

Correspondence: {bingyi,koert,andrearaujo}@google.com

Table 4. **Ablations for multi-granularity captions**, on ViT-g. Texts for 2 CLS are separated by / and for each CLS, texts are uniformly sampled from sources.

Text Strategy	Seg. ADE20k	Depth ↓ NYUv2	ImageNet KNN	IT Ret. COCO	0-shot Seg. ADE150
1 CLS (web, PaliGemma)	46.3	0.375	80.4	72.3	16.4
1 CLS (web, PaliGemma, Gemini)	46.7	0.366	81.2	74.4	17.7
2 CLS (web / PaliGemma)	48.3	0.370	**84.3**	70.0	17.1
2 CLS (web / PaliGemma, Gemini)	**49.1**	**0.354**	**84.3**	**76.2**	**18.1**

ratio from $75\% \rightarrow 0\%$ (compare rows (2-4)) . In iBOT++, we propose to take the approach in A) from distillation and apply it to pretraining. This naturally raises the question: Would applying B) by removing masking in iBOT++ pretraining also further enhance patch-text alignment? To explore this, we ablated the masking ratio during iBOT++ pretraining. Our experiments in Tab. 3 demonstrate that the answer is no: iBOT++ achieves its best performance with a 75% masking ratio. As a result, we adopt the 75% ratio setting as our final TIPSv2 training recipe. Our results confirm that Masked Image Modeling (MIM) remains critical for pretraining, consistently improving performance across image-only tasks and image-text tasks. We conclude that mask removal is only effective during distillation because the teacher model already provides the necessary strong local semantic understanding. This allows the student vision encoder to inherit the alignment without needing to learn it via the MIM objective.

A.3. Ablations on Multi-Granularity Captions

In Tab. 4, we present ablations to justify our strategy to utilize multiple captions, varying 1-2 [CLS] tokens and the assignment to [CLS] tokens between the 3 available captions (web, PaliGemma, Gemini). These ablations are conducted with the default TIPSv2 training setup, except that full EMA is used (instead of head-only EMA). We find the optimal strategy is the recipe in TIPSv2: alternating real/synthetic captions with the dual CLS setup.

A.4. Qualitative Comparisons to DINOv2 and v3

In order to further illustrate the comparison with self-supervised pretrained models, we provide additional PCA maps for DINOv2 (with registers) [2, 4] and DINOv3 [6]. Fig. 1 presents results on ViT-L models, which is the largest common size between these model families. Fig. 2 presents results on larger models: ViT-g for DINOv2 and TIPSv2, but ViT-7B for DINOv3. Note that the DINOv3 ViT-7B teacher model is trained with 6× more parameters and 15× more images than the TIPSv2 ViT-g teacher.

Comparing DINOv3 and TIPSv2, we can see that PCA maps for DINOv3 are smoother, whereas TIPSv2 maps are more granular. We note that DINOv3 enhances dense feature maps (on top of DINOv2) by means of a Gram anchoring loss that acts on patch correlations, introducing another

Figure 1. **PCA maps at ViT-L size.** Comparing the first 3 PCA components from the ViT-L models of DINOv2 (with registers), DINOv3, and TIPSv2. Images are forwarded at 1372 resolution for patch size 14 models (DINOv2 and TIPSv2) and at 1568 resolution for patch size 16 models (DINOv3). DINOv3 features appear smoother, but TIPSv2 features show more semantically focused features, e.g., TIPSv2 maps show all windows clustered together in row 1, and the eyes and leash are distinct on the dog in row 4.

teacher that is frozen from the past, along with a higher-resolution inference and down-sampling procedure. The dense feature maps of TIPSv2 also demonstrate comparable improvements in spatial coherence (on top of TIPS), but require much simpler changes to the training setup. On some images, this seems to demonstrate a more semantic focus for TIPSv2, compared to a more spatial focus in DINOv3.

For example, in the third row of Fig. 1 and Fig. 2, the backpacks appear semantically similar to the people wearing them in DINOv3 PCA, while they are clearly distinct from the people in TIPSv2 PCA. In the second row of Fig. 2, the ceiling tends to be underclustered in DINOv3, unable to distinguish lamps and other features; in contrast, TIPSv2 clearly segments them properly. In the first row of Fig. 1, the windows for DINOv3 PCA are distinctly colored, varying from foreground to background, but are all uniformly colored for TIPSv2 PCA.

Comparing against DINOv2, TIPSv2 PCA maps are overall much smoother and more spatially coherent, in contrast to DINOv2's noisy maps.

Figure 2. **PCA maps at ViT-g or ViT-7B size.** Comparing the first 3 PCA components from teacher models of DINOv2 (with registers, ViT-g), DINOv3 (ViT-7B), and TIPSv2 (ViT-g). Images are forwarded at 1372 resolution for patch size 14 models (DINOv2 and TIPSv2) and at 1568 resolution for patch size 16 models (DINOv3). As for the ViT-L PCA maps, DINOv3 features appear smoother, but TIPSv2 features capture more semantically focused details, e.g., notice the ceiling details in row 2 or the different colors contrasting people and backpacks in row 3.

A.5. Qualitative Analysis: iBOT++ vs iBOT

As detailed in Section 3.3, iBOT++ significantly enhances patch-text alignment during pretraining. This improvement is quantified in Table 4 (from the main paper), where a substantial performance gain is directly observed in the zero-shot segmentation metrics when comparing iBOT baseline (first row) to iBOT++ (second row). To further illustrate this enhancement, Fig. 3 provides qualitative visualizations of the zero-shot segmentation results, which correspond directly to the first and second rows of Table 4 (from the main paper). These visualizations confirm that iBOT++ yields significantly cleaner segmentation maps.

A.6. Zero-shot Segmentation with SigLIPv2

We showed that smaller models in the TIPS [3] family outperform larger ones in the task of zero-shot image segmentation, which motivated our investigations and contributions towards TIPSv2. In this section, we provide evidence of a similar effect also happening in the SigLIP2 family. Tab. 5 presents zero-shot segmentation results for three model sizes, where the smallest one outperforms the larger versions in two evaluations, and the SO size model

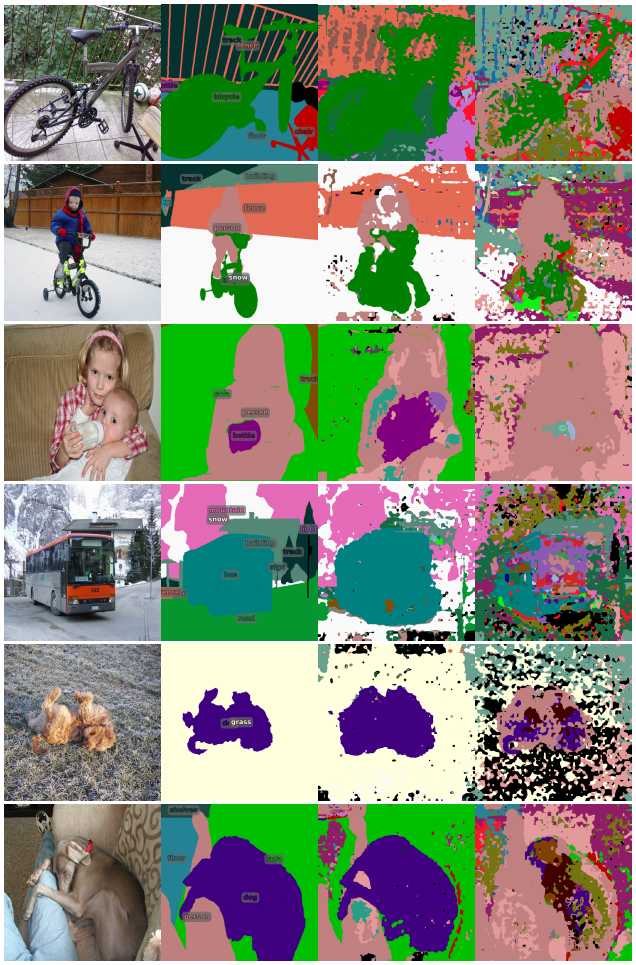

Figure 3. **Zero-shot segmentation visualization.** Comparing results with the iBOT loss (used in TIPS) vs the iBOT++ loss (used in TIPSv2), where classes are predicted directly by finding the closest text embedding to each image patch token, without any post-processing. Compared to baseline iBOT, iBOT++ achieves significantly improved patch-text alignment, as part of the TIPSv2 recipe.

Table 5. **Zero-shot segmentation for SigLIP2 models.** Similarly to TIPS, the smaller models tend to outperform larger ones in the same model family.

Model	Zero-shot Segmentation (mIoU) ↑		
	PC60	VOC21	ADE150
SigLIP2 B/16	**22.6**	25.8	**16.4**
SigLIP2 SO/14	19.6	**26.8**	15.6
SigLIP2 g/16	17.2	25.7	13.9

wins in the third; notably, the largest ViT-g model presents the worst performance in all three evaluations. Note that the smallest ViT-B size model is distilled via active data curation from a larger SO model in the family. Similarly to the TIPS case, these results indicate that the smaller models surprisingly outperform larger pretrained ones.

Table 6. **Evaluations for all TIPSv2 model variants.** We report four distinct TIPSv2 model sizes (ViT-B, ViT-L, ViT-g, and SO-400m). Only the ViT-g model is pretrained; the other sizes (ViT-B, ViT-L, and SO-400m) are distilled from the ViT-g teacher. Best performance for each task is highlighted.

Model	Segmentation ↑ PASCAL	Depth ↓ NYUv2	Normals ↓ NYUv2	KNN ↑ ImageNet	I→T Ret. ↑ Flickr	T→I Ret. ↑ Flickr	Zero-shot Seg. ↑ ADE150
TIPSv2 g/14	85.1	**0.334**	**21.7**	**83.7**	95.1	**85.9**	17.8
TIPSv2 SO/14	**85.2**	0.339	**21.7**	82.8	94.8	84.0	23.3
TIPSv2 L/14	85.1	0.339	21.9	82.5	**95.4**	83.3	**24.7**
TIPSv2 B/14	84.0	0.374	23.2	79.8	92.6	80.0	17.4

Table 7. **Number of parameters for all TIPSv2 model variants.** We release 4 different model variants. For B, L and g model sizes, we use a fixed number of 12 layers in the text encoder; for the SO size, we use the same number of layers in both the image and text encoders.

Model	Image # Params	Text # Params	Total # Params
TIPSv2-B/14	86.3M	109.6M	195.9M
TIPSv2-L/14	304.0M	183.9M	487.9M
TIPSv2-SO/14	413.3M	448.3M	861.7M
TIPSv2-g/14	1.1B	389.1M	1.5B

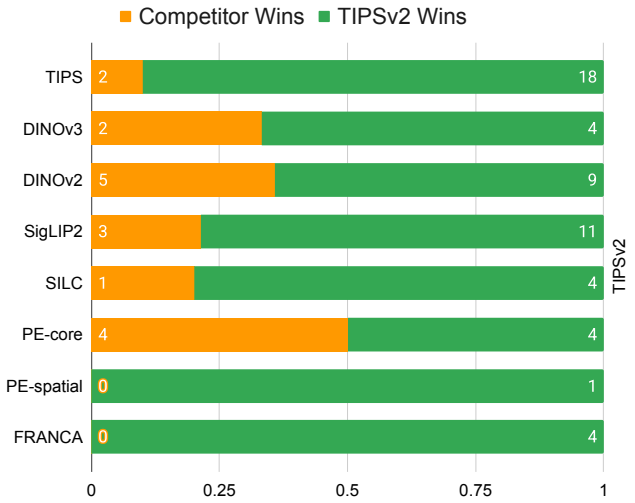

Figure 4. **Comparison of TIPSv2 against recent vision encoders.** The chart illustrates the percentage of shared evaluation benchmarks where TIPSv2 secures the best result (green) compared head-to-head to other individual leading models (orange). TIPSv2 demonstrates a winning record on the majority of shared tasks. The integer displayed on the chart represents the count of metrics on which each respective model achieves the best result (i.e., the number of wins). For each comparison between individual vision encoders, we use the largest comparable model size (ViT-g against ViT-g, or ViT-g against ViT-G, or ViT-L against ViT-L).

A.7. TIPSv2 Family

The TIPSv2 model family includes variants with different backbone sizes: ViT-B, ViT-L, ViT-g and SO-400m [1] (referred to as SO hereafter). The ViT-g model is directly pretrained, while the smaller ViT-B, ViT-L, and SO variants are obtained via the patch-level distillation strategy detailed in Sections 3.1 and 3.2, using ViT-g as the teacher model. For text encoder scaling, we follow [3] and adopt the same transformer parameterization as the image encoder but fix the number of layers at 12 (except for the SO variant, which maintains its standard layer count).

Our studies demonstrate that the TIPSv2 family exhibits strong performance across all image and image-text benchmarks, as summarized in Tab. 6. The parameter count for each model is detailed in Tab. 7.

A.8. Performance of TIPSv2 against Competitors

To gain a comprehensive understanding of the performance of our new method compared to other recent vision encoders, we quantified the number of evaluations where TIPSv2 outperforms (or underperforms) each competitor method head-to-head, based on reported scores in the tables in the main paper. Looking only at the evaluations the models share, TIPSv2 generally comes out on top in most cases, as shown in Fig. 4. This summary confirms the strong results from TIPSv2 overall.

A.9. Additional Implementation Details

Our implementation closely follows that of TIPS. The loss components are weighted by $\mathcal{L} = \mathcal{L}_{\text{CLIP}} + \alpha\mathcal{L}_{\text{DINO}} + \beta\mathcal{L}_{\text{iBOT}}$, where $\alpha = 1.0$, $\beta = 2.0$. $\mathcal{L}_{\text{CLIP}}$ is the averaged result of contrastive losses from the two captions. We use the Adafactor [5] optimizer, and projection heads are balanced from collapse with EMA centering and sharpening.

References

[1] Ibrahim Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting vit in shape: Scaling laws for compute-optimal model design, 2024. 4

[2] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. Vision Transformers Need Registers. In *Proc. ICLR*, 2024. 2

[3] K. Maninis, K. Chen, S. Ghosh, A. Karpur, K. Chen, Y. Xia, B. Cao, D. Salz, G. Han, J. Dlabal, D. Gnanapragasam, M. Seyedhosseini, H. Zhou, and A. Araujo. TIPS: Text-Image Pretraining with Spatial Awareness. In *Proc. ICLR*, 2025. 1, 3, 4

[4] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P. Huang, H. Xu, V. Sharma, S. Li, W. Galuba,

M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. 2

[5] N. Shazeer and M. Stern. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. In *Proc. ICML*, 2018. 4

[6] O. Siméoni, H. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolan, J. Brandt, C. Couprie, J. Mairal, H. Jégou, P. Labatut, and P. Bojanowski. DINOv3. *arXiv:2508.10104*, 2025. 2