Exploring Visual Pretraining for Learning Language Intelligence
Supplementary Material

A. Outline

The supplementary material is structured as follows.

- **Section B (MAPLE Pseudo Code).** We present algorithmic descriptions of MAPLE, including page–to–latent conversion, the LLM-based latent prediction, and the joint training loop, and summarise the main notation used in the pseudo code.
- **Section D (Loss–performance correlation on Olympiad).** We extend the main-text analysis of loss–performance correlation to OlympiadBench [3], reporting per-backbone scatter plots for both image and text losses and discussing the differences from MATH-500 [4].
- **Section E (Evaluation Results on Multidisciplinary Knowledge).** We extend the main-text math-centric evaluation to broader multidisciplinary knowledge benchmarks, including MMLU and MMLU-pro, and further compare MAPLE with the corresponding vision-language model.
- **Section F (Case on model scale and image loss).** We provide qualitative reconstructions under different model scales and image loss designs, illustrating how larger backbones and the combined diffusion+MSE loss improves page readability and visual fidelity.
- **Section G (Evolution of visual–text embedding geometry).** We analyse how projected patch embeddings and token embeddings co-organise during pretraining via t-SNE and nearest-neighbour statistics, showing that MAPLE progressively tightens math-related visual–text clusters and induces a more compact, semantically aligned latent space.
- **Section H (Additional Implementation Details)** We describe optimization hyperparameters, training schedules for baselines, and MAPLE.

B. MAPLE Pseudo Code

This section summarises our visual next-patch pretraining pipeline in algorithmic form. For clarity, we refer to our method as MAPLE, which uses document images as the primary pretraining signal and treats text CPT as a standard, auxiliary path.

Algorithm 1 describes how a PDF page is converted into a sequence of sparse foreground latents in reading order. A rendered page $\mathcal{I}$ is passed through a frozen VAE encoder f_{VAE} to obtain a latent grid $\mathcal{Z}$, then a masked autoregressive encoder f_{MAR} produces a foreground-only sequence $\mathcal{U} = \{u_i\}$ in raster order, together with positional encodings implicitly defined by $\mathrm{PE}_{1\mathrm{D}}$ and $\mathrm{PE}_{2\mathrm{D}}$. These latents serve as the visual tokens to be modelled autoregressively.

Algorithm 2 gives the forward pass of MAPLE on a visual sequence. The foreground latents $\mathcal{U}$ are projected into the LLM hidden space as $\tilde{\mathcal{U}}$ and fed into the shared LLM f_θ under a causal mask over image positions. The LLM outputs latent hypotheses $\hat{\mathcal{U}}$, which condition a MAR decoder g_{MAR} and a frozen VAE decoder g_{VAE} to reconstruct the page $\hat{\mathcal{I}}$. In parallel (not shown in detail), the same LLM runs a conventional next-token objective on text-only batches.

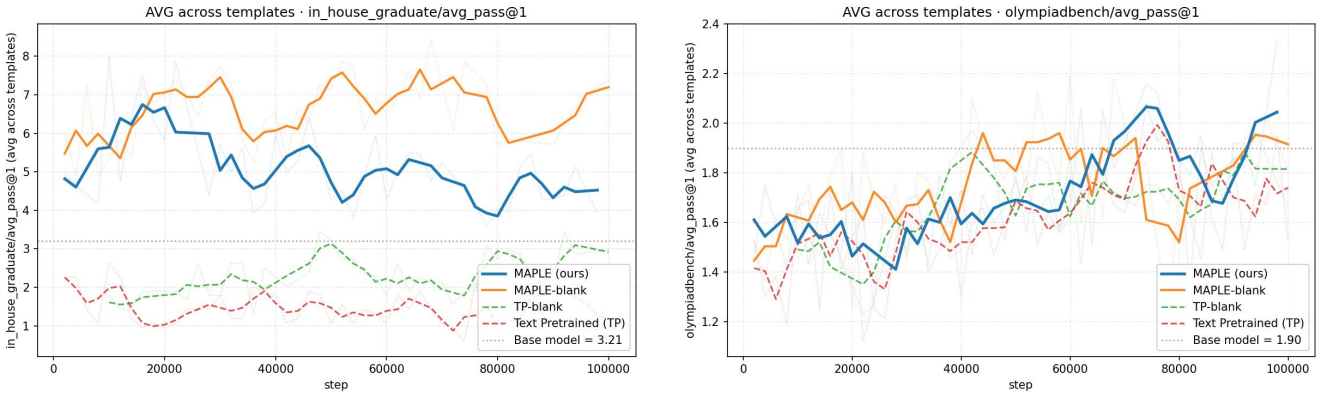
Algorithm 3 summarises the joint training loop. Each step samples either an image batch or a text batch according to the mixing ratio ρ. Image batches are trained with a diffusion-style loss on latents and an MSE loss on pixels, while text batches are trained with a standard cross-entropy loss. The joint objective

$$\mathcal{L} = \lambda_{\text{text}}\,\mathcal{L}_{\text{CE}} + \lambda_{\text{diff}}\,\mathcal{L}_{\text{diff}} + \lambda_{\text{pix}}\,\mathcal{L}_{\text{MSE}}$$

encourages the LLM to act as a visual latent predictor while preserving its language modelling ability.

Notation. For convenience, we summarise the main symbols used in the pseudo code:

- $\mathcal{I}$: rendered PDF page (RGB image).
- $\mathcal{Z}$: VAE latent grid of $\mathcal{I}$.
- $\mathcal{U} = \{u_i\}$: sequence of sparse foreground latents in reading order.
- e_i^{pos}: positional encoding for latent u_i (from $\mathrm{PE}_{1\mathrm{D}}$ and $\mathrm{PE}_{2\mathrm{D}}$).
- $\tilde{\mathcal{U}} = \{\tilde{u}_i\}$: projected latents in the LLM hidden space.
- $\hat{\mathcal{H}} = \{\hat{h}_i\}$: LLM hypothesis.
- $\hat{\mathcal{U}} = \{\hat{u}_i\}$: LLM-refined latent.
- $\hat{\mathcal{Z}}$: reconstructed latent grid from the MAR decoder.



(a) Inhouse-graduate (b) OlympiadBench

Figure C1. **Training dynamics on additional math benchmarks.** With InternLM2-1.8B [1] as a clean base, we compare MAPLE, MAPLE-blank, TP-blank, and TP on (a) an in-house graduate-level math benchmark and (b) OlympiadBench. MAPLE-blank tracks TP-blank closely, while MAPLE achieves the highest final accuracy on both datasets.

- $\hat{\mathcal{I}}$: reconstructed page from MAR + VAE decoders.
- f_{VAE}, g_{VAE}: VAE encoder and decoder (frozen in Stage 2).
- f_{MAR}, g_{MAR}: masked autoregressive encoder and decoder.
- f_θ: shared autoregressive LLM.
- W_{in}, W_{out}: linear maps between visual latent space and LLM hidden space.
- $\mathcal{L}_{\text{diff}}$, $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{CE}}$: diffusion-style latent loss, pixel MSE loss, and text cross-entropy loss.
- θ: trainable parameters of the whole model (LLM and visual modules except the frozen VAE).

Together, these algorithms highlight the core idea of MAPLE: use a single LLM as a causal predictor over document latents, trained end-to-end with MAR and VAE modules, so that visual pretraining on pages can directly enhance the language intelligence of the backbone model.

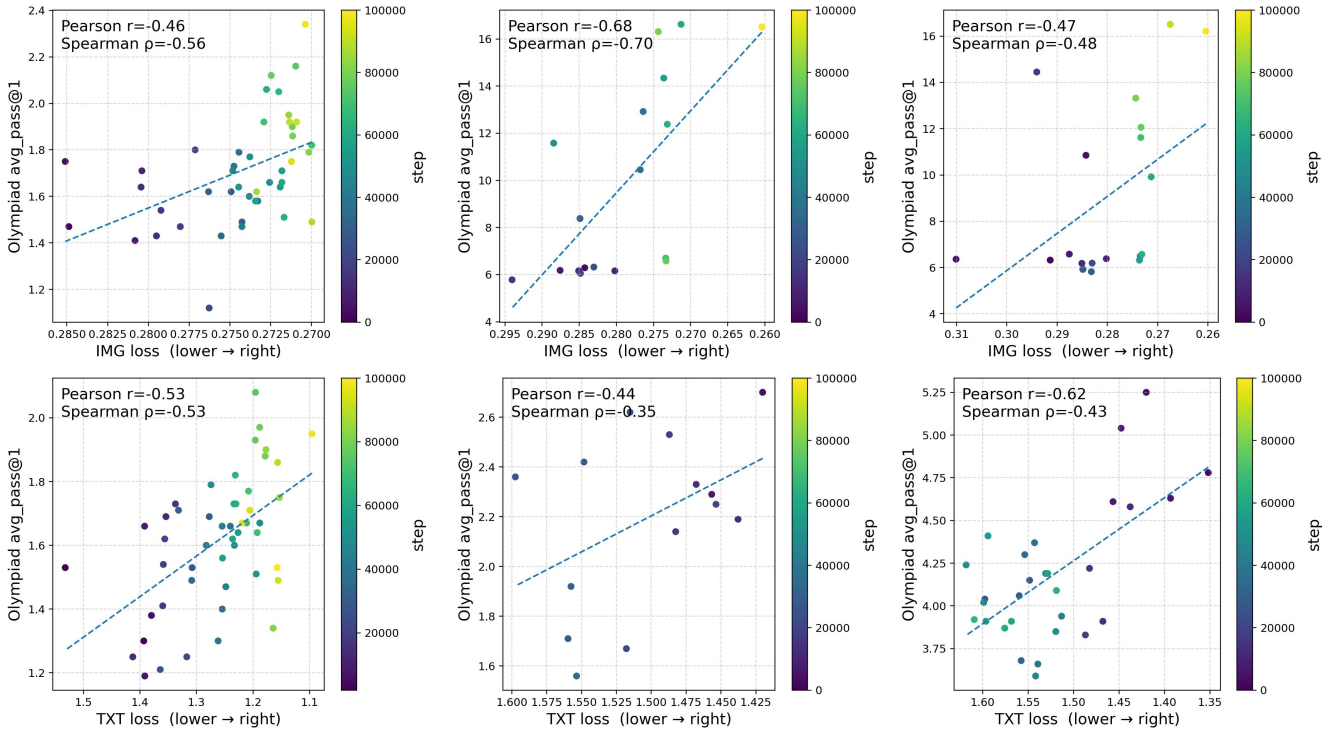
C. Training Dynamics on Other Benchmarks

We further track training dynamics on two harder math benchmarks using InternLM2-1.8B [1] as a clean base: an in-house *graduate-level* math set and OlympiadBench. Across both datasets, MAPLE consistently outperforms pure text pretraining (TP) and TP-blank, while MAPLE-blank even approaches MAPLE on the graduate benchmark. A likely reason is that the graduate corpus is much smaller than the general-domain math data, so repeated exposure makes the blank-text setting relatively strong, whereas MAPLE additionally exploits page images and equations and thus avoids overfitting or degradation.

Both benchmarks remain challenging for such a light base model without any SFT-style instruction or chain-of-thought tuning, so the performance ceiling is reached quickly. Nevertheless, these results show that visual-document pretraining is competitive when the information content is matched, and yields clear gains on graduate and Olympiad-style problems when real PDF pages with figures and formulas are available.

D. Loss–performance correlation on Olympiad

Figure D2 reports the same analysis as Figure 4 but on OlympiadBench [3]. For each backbone, we scatter Olympiad Pass@1 against our image loss (top) and the text loss of the text-pretraining baseline (bottom). Overall, both losses still show a clear positive trend: runs with lower training loss tend to achieve higher Olympiad accuracy. However, the correlations are weaker and noisier than on MATH-500 [4], which matches the higher difficulty and stochasticity of Olympiad-style problems. This suggests that Olympiad performance is still predictable from both visual and text losses, but is more sensitive to optimization variance and sample complexity.



(a) InternLM2-1.8B (b) Qwen2.5-1.5B (c) Llama3.1-8B

Figure D2. **Loss–performance correlation on OlympiadBench.** For three backbones, (a) InternLM2-1.8B, (b) Qwen2.5-1.5B (early segment), and (c) Llama3.1-8B (early segment), we plot OlympiadBench Pass@1 against two training losses: the image-side loss of our visual pretraining L_{img} (top row) and the text cross-entropy loss from the corresponding text-pretraining baseline (bottom row). The x-axis is inverted (lower loss on the right). Due to evaluation cost, for Qwen2.5-1.5B and Llama3.1-8B we report a contiguous set of early checkpoints (the same range used in the main results), which is sufficient to reveal the correlation trend, especially in the early phase. Points are colored by training step, and the dashed line shows a least-squares fit. Across all models, lower losses tend to align with higher Olympiad scores, although the correlation is noisier than on MATH-500.

Model	**SFT Pass@1**				
	MATH	**Olympiad**	**AIME-24**	**MMLU**	**MMLU-pro**
InternLM2-1.8B (base)	15.6	6.85	3.33	31.2	11.7
InternVL2-2B-instruct [2]	10.2	2.48	0	38.8	15.95
MAPLE-InternLM2-1.8B	21.8	9.38	10	45.4	16.9

Table D1. Evaluation results across a broad range of benchmarks for vision-language models.

E. Evaluation Results on Multidisciplinary Knowledge

We focus on math reasoning in the main text because it provides a clean, minimal, closed-loop testbed for validating MAPLE. Math benchmarks require strong reasoning and offer a controlled setting to show the core effect of visually grounded pretraining on the language model.

To broaden the evaluation, we further report results on multidisciplinary knowledge benchmarks (with 100B extra multidisciplinary data pretraining), including MMLU and MMLU-pro (Table D1). We also compare with the corresponding vision-language model, InternVL2-2B-instruct, to verify that the gain comes from the MAPLE pretraining strategy rather than simply using a generic VL model. As shown in Table D1, MAPLE-InternLM2-1.8B consistently improves over the InternLM2-1.8B base model on both general knowledge and math-related benchmarks, and also outperforms InternVL2-2B-

instruct on most tasks.

Overall, these results extend the evidence for MAPLE beyond the math-focused setting in the main text. They suggest that MAPLE is already effective at the pretraining stage, improving not only mathematical reasoning but also broader multidisciplinary knowledge. We do not include additional code benchmarks, since code generation is mainly text-native and is less related to document-style visual structure, which is not the primary target of MAPLE.

F. Case on Model Scale and Image Loss

Section 4.3 in main text showed that our image loss is positively correlated with downstream language performance on math benchmarks. Figure H3 complements this result from a case-based perspective. We select InternLM2-1.8B and Llama3.1-8B, the two backbones that achieve the best scores in Section 4.2 under our method, and compare their reconstructed pages under different image-loss choices. Comparing (a) to (b), using the stronger Llama3.1-8B backbone markedly improves the readability of text and equations even with the same $\mathcal{L}_{\text{diff}}$ loss. Comparing (a)/(b) with (c)/(d), adding the MSE term to form the combo loss $\mathcal{L}_{\text{diff}}+\mathcal{L}_{\text{MSE}}$ further sharpens characters, stabilizes line geometry, and reduces artifacts. The best visual quality is obtained with the 8B model and the combo loss (d), supporting our claim that both LLM capacity and the carefully designed IMG loss are crucial for capturing rich document knowledge in the joint model.

G. Evolution of Visual–Text Embedding Geometry

Understanding how MAPLE reshapes the shared visual–text space is key to our representation analysis [6, 7]. Our study is related to prior work that probes cross-modal alignment via nearest-neighbor retrieval in a joint embedding space [5, 8, 9]. Unlike these methods, which mostly align images with full sentences, we focus on how document *patch* embeddings move closer to math-related tokens in the LLM vocabulary as training progresses.

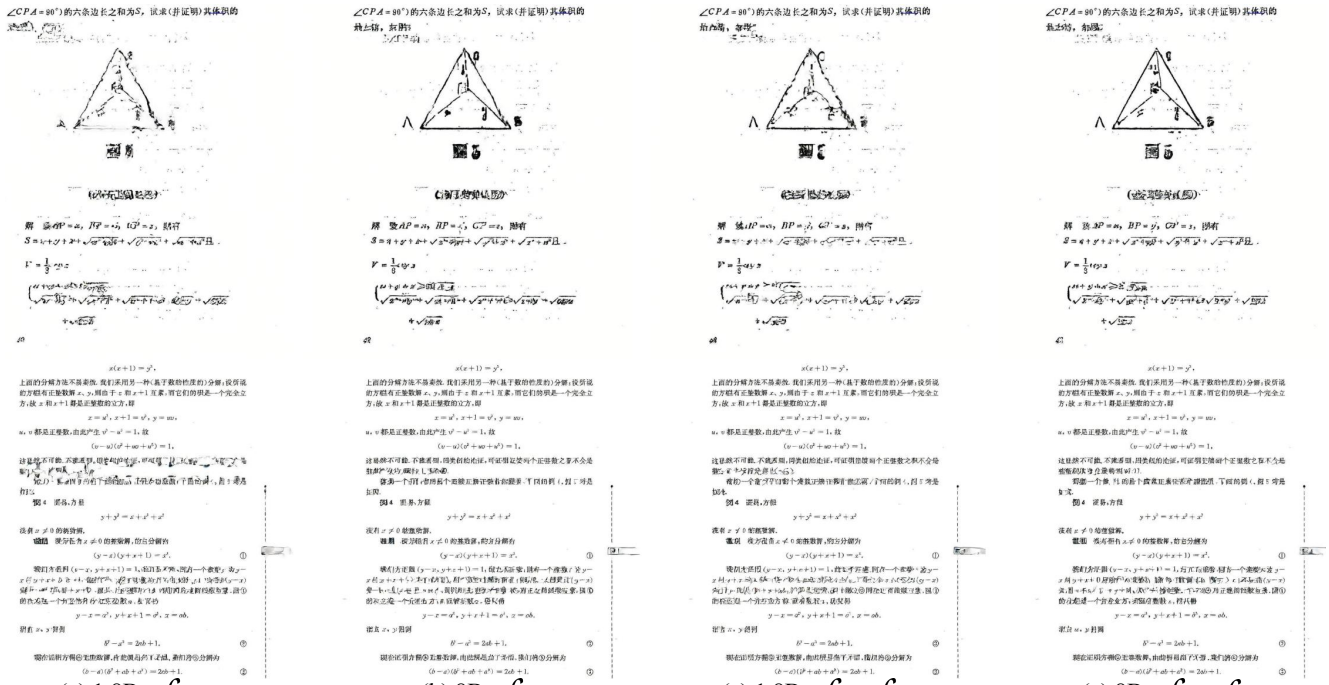
We probe the embeddings *before* entering the LLM. For each checkpoint, we take the projected patch embeddings from document pages and a uniformly sampled set of 20K vocabulary token embeddings. We then: (1) compute, for every patch, its nearest text token in cosine similarity and plot the similarity histogram, and (2) run t-SNE [10] over all patch embeddings, colouring each point by the identity of its nearest text token (Figure H4).

In the base model (Figure H4 a), the most similar tokens have little relation to mathematics, and the similarity histogram is multi-modal and noisy. This suggests that visual patches are only weakly organised around semantic tokens. After Stage 1 warm-up (Figure H4 b–c), the patch manifold becomes more structured and separable. Nearest neighbours now contain more math-related tokens, and the similarity histogram becomes more concentrated, indicating that the visual encoder learns a cleaner, math-aware latent space.

Further training with MAPLE (Figure H4 d–f) strengthens this effect. Clusters associated with digits, operators, brackets, and problem-related keywords become tighter, and the similarity histogram approaches a unimodal, near-Gaussian shape. This implies that (1) visually text-like patches are mapped to more distinct, well-separated regions, and (2) high-frequency math tokens in the CPT data pull nearby visual patches into a compact math-centric subspace. Overall, these observations support our claim that MAPLE encourages a shared, semantically aligned representation in which document visual signals and mathematical text occupy a coherent geometry.

H. Additional Implementation Details.

Optimization is performed with bfloat16 mixed precision and dynamic loss scaling via `AmpOptimWrapper`. The *text pretraining* baselines are trained with an identical optimizer and schedule on the same corpus but with no image branches, while MAPLE mixes image and text batches with ratio $\rho = \frac{N}{N+M}$ as described in §3. For the instruction-tuned models, we run a subsequent SFT stage starting from the corresponding base (*text pretraining* or MAPLE) checkpoint, using standard autoregressive cross-entropy on supervised dialogue/math data and the same optimizer family and LR schedule but a much shorter training horizon.



(a) 1.8B + $\mathcal{L}_{\text{diff}}$ (b) 8B + $\mathcal{L}_{\text{diff}}$ (c) 1.8B + $\mathcal{L}_{\text{diff}}\mathcal{L}_{\text{MSE}}$ (c) 8B + $\mathcal{L}_{\text{diff}}\mathcal{L}_{\text{MSE}}$

Figure H3. **Qualitative effect of model scale and image loss design.** We visualise reconstructed pages for two backbones and two image-loss settings. From left to right: (a) InternLM2-1.8B with $\mathcal{L}_{\text{diff}}$ only, (b) Llama3.1-8B with $\mathcal{L}_{\text{diff}}$ only, (c) InternLM2-1.8B with $\mathcal{L}_{\text{diff}}+\mathcal{L}_{\text{MSE}}$, (d) Llama3.1-8B with $\mathcal{L}_{\text{diff}}+\mathcal{L}_{\text{MSE}}$. Larger LLM capacity and the combined diffusion+MSE image loss both lead to sharper strokes, clearer formulas, and cleaner page layout, in line with the quantitative loss–performance correlation observed in Section 4.3.

Algorithm 1 MAPLE: Page $\rightarrow$ Sparse Foreground Latents

Input: Document page image $\mathcal{I}$

Modules:
- f_{VAE}: VAE encoder (frozen)
- f_{MAR}: MAR encoder (masked, raster order)
- PE_{1D}, PE_{2D}: positional encoders
- W_{in}: linear map from visual latents to LLM space

Variables:
- $\mathcal{Z}$: latent grid
- $\mathcal{U} = \{u_i\}$: sparse foreground latents
- $\tilde{\mathcal{U}} = \{\tilde{u}_i\}$: projected latents

procedure PAGETOLATENTS($\mathcal{I}$)
- $\mathcal{Z} \leftarrow f_{\text{VAE}}(\mathcal{I})$
- $\mathcal{U} \leftarrow f_{\text{MAR}}(\mathcal{Z})$ ▷ foreground-only, raster order
- **for** each latent $u_i \in \mathcal{U}$ **do**
  - compute global index t_i and normalized coord $(x_i/W, y_i/H)$
  - $e_i^{\text{pos}} \leftarrow \text{PE}_{\text{1D}}(t_i) + \text{PE}_{\text{2D}}(x_i/W, y_i/H)$
  - $\tilde{u}_i \leftarrow W_{\text{in}} u_i + e_i^{\text{pos}}$
- **end for**
- **return** $\mathcal{U}, \tilde{\mathcal{U}}$

end procedure

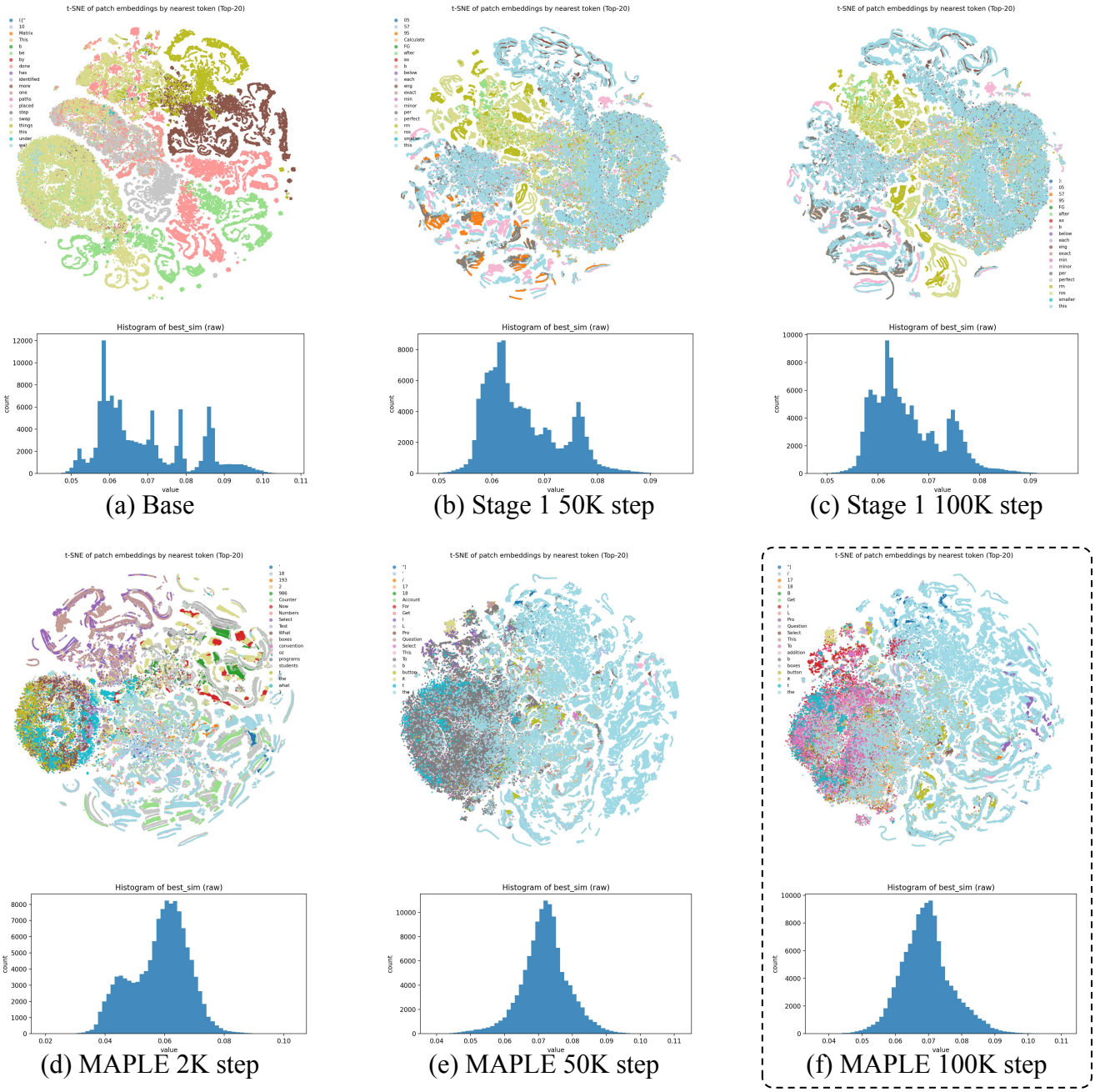


Figure H4. **Evolution of visual–text embedding geometry during pretraining.** For each training stage, we project foreground patch embeddings (before entering the LLM) and a uniformly sampled set of 20K vocabulary token embeddings into a shared space. Top: t-SNE visualisation of patch embeddings coloured by their nearest text-token neighbours. Bottom: histogram of cosine similarities between each patch and its top-1 text neighbour. From left to right: (a) Base model, (b) Stage 1 at 50K steps, (c) Stage 1 at 100K steps, (d) MAPLE at 2K steps, (e) MAPLE at 50K steps, (f) MAPLE at 100K steps.

Algorithm 2 MAPLE: LLM Hypotheses for Page Reconstruction

Input: Foreground latents $\mathcal{U}$, projected latents $\tilde{\mathcal{U}}$

Modules:
  f_θ: LLM with causal attention over image positions
  W_{out}: linear map from LLM space to visual latent space
  g_{MAR}: MAR decoder
  g_{VAE}: VAE decoder (frozen)
Variables:
  $\mathcal{H} = \{h_i\}$: LLM hidden states
  $\hat{\mathcal{U}} = \{\hat{u}_i\}$: refined latent hypotheses
  $\hat{\mathcal{Z}}$: reconstructed latent grid
  $\hat{\mathcal{I}}$: reconstructed image
procedure HYPOTHESIZEANDRECONSTRUCT($\mathcal{U}, \tilde{\mathcal{U}}$)
    build causal mask (raster order)
    $\mathcal{H} \leftarrow f_\theta(\tilde{\mathcal{U}})$
    for each position i **do**
      $\hat{u}_i \leftarrow u_i + W_{\text{out}} h_i$
    end for
    $\hat{\mathcal{Z}} \leftarrow g_{\text{MAR}}(\hat{\mathcal{U}})$
    $\hat{\mathcal{I}} \leftarrow g_{\text{VAE}}(\hat{\mathcal{Z}})$
    return $\hat{\mathcal{U}}$, $\hat{\mathcal{Z}}$, $\hat{\mathcal{I}}$
end procedure

Algorithm 3 MAPLE: One Joint Training Step

Input: Image batch $\{\mathcal{I}\}$, text batch $\{\mathcal{T}\}$

Hyper-parameters:
  $\lambda_{\text{text}}, \lambda_{\text{diff}}, \lambda_{\text{pix}}$
  mixing ratio ρ ▷ prob. of sampling an image batch
Losses:
  $\mathcal{L}_{\text{diff}}$: diffusion / denoising loss on latents
  $\mathcal{L}_{\text{MSE}}$: pixel MSE between $\mathcal{I}$ and $\hat{\mathcal{I}}$
  $\mathcal{L}_{\text{CE}}$: LM cross-entropy on text tokens
procedure TRAINSTEP
    if sampled image step with prob. ρ **then**
      sample a mini-batch of pages $\{\mathcal{I}\}$
      for each $\mathcal{I}$ in batch **do**
        $\mathcal{U}, \tilde{\mathcal{U}} \leftarrow$ PAGETOLATENTS($\mathcal{I}$)
        $\hat{\mathcal{U}}, \hat{\mathcal{Z}}, \hat{\mathcal{I}} \leftarrow$ HYPOTHESIZEANDRECONSTRUCT($\mathcal{U}, \tilde{\mathcal{U}}$)
      end for
      $\mathcal{L}_{\text{diff}} \leftarrow \text{DiffusionLoss}(\hat{\mathcal{U}}, \mathcal{U})$
      $\mathcal{L}_{\text{MSE}} \leftarrow \|\mathcal{I} - \hat{\mathcal{I}}\|_2^2$
      $\mathcal{L}_{\text{CE}} \leftarrow 0$
    else
      sample a mini-batch of text $\{\mathcal{T}\}$
      embed $\{\mathcal{T}\}$ and run LLM with text-only causal mask
      $\mathcal{L}_{\text{CE}} \leftarrow \text{CrossEntropy}(\text{text logits}, \{\mathcal{T}\})$
      $\mathcal{L}_{\text{diff}} \leftarrow 0, \quad \mathcal{L}_{\text{MSE}} \leftarrow 0$
    end if
    $\mathcal{L} \leftarrow \lambda_{\text{text}}\mathcal{L}_{\text{CE}} + \lambda_{\text{diff}}\mathcal{L}_{\text{diff}} + \lambda_{\text{pix}}\mathcal{L}_{\text{MSE}}$
    update f_θ, f_{MAR}, g_{MAR}, W_{in}, W_{out} by $\nabla\mathcal{L}$
end procedure

References

[1] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report, 2024.

[2] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101, 2024.

[3] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

[4] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

[5] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. pages 19730–19742. PMLR, 2023.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[7] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*, 2017.

[8] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.