Dynamic resolution can be composed of two native resolutions. For example, Gundam mode consists of n×640×640 tiles (local views) and a 1024×1024 global view. The tiling method following InternVL2.0 [8]. Supporting dynamic resolution is mainly for application considerations, especially for ultra-high-resolution inputs (such as newspaper images). Tiling is a form of secondary window attention that can effectively reduce activation memory further. It's worth noting that due to our relatively large native resolutions, images won't be fragmented too much under dynamic resolution (the number of tiles is controlled within the range of 2 to 9). The vision token number output by the DeepEncoder under Gundam mode is: $n \times 100 + 256$, where n is the number of tiles. For images with both width and height smaller than 640, n is set to 0, i.e., Gundam mode will degrade to Base mode.

Gundam mode is trained together with the four native resolution modes to achieve the goal of one model supporting multiple resolutions. Note that Gundam-master mode (1024×1024 local views+1280×1280 global view) is obtained through continued training on a trained DeepSeek-OCR model. This is mainly for load balancing, as Gundam-master's resolution is too large and training it together would slow down the overall training speed.

3.3. The MoE Decoder

Our decoder uses the DeepSeekMoE [19, 20], specifically DeepSeek-3B-MoE. During inference, the model activates 6 out of 64 routed experts and 2 shared experts, with about 570M activated parameters. The 3B DeepSeekMoE is very suitable for domain-centric (OCR for us) VLM research, as it obtains the expressive capability of a 3B model while enjoying the inference efficiency of a 500M small model.

The decoder reconstructs the original text representation from the compressed latent vision tokens of DeepEncoder as:

$$f_{\text{dec}} : \mathbb{R}^{n \times d_{\text{latent}}} \rightarrow \mathbb{R}^{N \times d_{\text{text}}}; \quad \hat{\mathbf{X}} = f_{\text{dec}}(\mathbf{Z}) \quad \text{where } n \leq N \tag{2}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times d_{\text{latent}}}$ are the compressed latent(vision) tokens from DeepEncoder and $\hat{\mathbf{X}} \in \mathbb{R}^{N \times d_{\text{text}}}$ is the reconstructed text representation. The function f_{dec} represents a non-linear mapping that can be effectively learned by compact language models through OCR-style training. It is reasonable to conjecture that LLMs, through specialized pretraining optimization, would demonstrate more natural integration of such capabilities.

3.4. Data Engine

We constructe complex and diverse training data for DeepSeek-OCR, including OCR 1.0 data, which mainly consists of traditional OCR tasks such as scene image OCR and document OCR; OCR 2.0 data, which mainly includes parsing tasks for complex artificial images, such as common charts, chemical formulas, and plane geometry parsing data; General vision data, which is mainly used to inject certain general image understanding capabilities into DeepSeek-OCR and preserve the general vision interface.

3.4.1. OCR 1.0 data

Document data is the top priority for DeepSeek-OCR. We collect 30M pages of diverse PDF data covering about 100 languages from the Internet, with Chinese and English accounting for approximately 25M and other languages accounting for 5M. For this data, we create two types of ground truth: coarse annotations and fine annotations. Coarse annotations are extracted