WeMMU: Enhanced Bridging of Vision-Language Models and Diffusion Models via Noisy Query Tokens

Supplementary Material

A. Model Architecture Details

A.1. Position MLP Design

The Position MLP module is engineered to dynamically encode spatial information for variable-sized feature maps and subsequently transform these spatially-aware features through a high-capacity gated network. The architecture comprises two primary stages: dynamic position encoding extraction and a gated MLP transformation.

To accommodate varying input image resolutions without requiring interpolation of position embeddings, which can degrade performance, we employ a dynamic cropping strategy. We first initialize a large, learnable position embedding matrix, $\mathrm{P} \in \mathbb{R}^{N*N*D_{pos}}$, using a "torch.nn.Embedding layer". Here, $N = 74$ is chosen to be larger than the maximum expected number of patches along any single dimension, and D_{pos} is the dimension of the position embeddings. Given an input feature map from the vision encoder, $\mathrm{F}_{in} \in \mathbb{R}^{H^{'}*W^{'}*C_{in}}$, where $H^{'}$ and $W^{'}$ are the number of patches along the height and width respectively, and $C_{in} = 2048$. We extract a corresponding sub-matrix of position embeddings, $\mathrm{P}_{crop} \in \mathbb{R}^{H^{'}*W^{'}*C_{in}}$, by cropping the central $H^{'} * W^{'}$ region from the larger matrix P. This ensures that the model always utilizes the most well-trained central portion of the positional space. The extracted position embeddings are then fused with the input features via element-wise addition:

$$F_{fused} = F_{in} + P_{crop} \tag{1}$$

This design provides flexibility in handling diverse aspect ratios and resolutions while maintaining a consistent and stable representation of spatial locality.

Following the spatial information fusion, the resulting tensor F_{fused} is processed by a Gated Multi-Layer Perceptron (MLP). This structure allows the network to modulate the feature representation more effectively than a standard MLP. The input features (with dimension $C_{in} = 2048$) are first projected in parallel into a higher-dimensional space ($C_{hidden} = 16384$) by two separate linear layers, an up-projection layer (W_{up}) and a gating layer (W_{gate}). The output of the up-projection layer is used to multiplicatively gate the output of the gating layer, followed by an activation function (σ). The process can be formulated as:

$$\mathrm{H} = \sigma(Linear_{gate}(\mathrm{F}_{\text{fused}})) \odot Linear_{up}(\mathrm{F}_{\text{fused}}) \tag{2}$$

where $\odot$ denotes element-wise multiplication. Subsequently, a final down-projection linear layer (W_{down}) maps the intermediate representation $\mathrm{H} \in \mathbb{R}^{H^{'}*W^{'}*C_{hidden}}$ back to the desired output dimension ($C_{out} = 2304$):

$$\mathrm{F}_{out} = Linear_{down}(H) \tag{3}$$

This gated mechanism empowers the model to control the information flow, enabling a more nuanced transformation of the spatially-enriched features.

A.2. Simple ViT Design

The Simple Vision Transformer (ViT) was developed as one of our initial exploratory methods for injecting VAE features into the Vision-Language Model (VLM). This architecture was conceived to investigate whether a dedicated, parameter-heavy module for processing these features could accelerate model training and improve convergence rates. Unlike a traditional ViT that operates on image patches, our Simple ViT module directly processes latent features extracted from a VAE. The architecture bypasses any image-patching and patch-embedding stages. The operational flow is as follows:

Input Projection. The input VAE features, denoted as F_{vae}, are first passed through a linear projection layer. This layer maps the features from their original VAE latent dimension to the Simple ViT's internal working dimension of 2048, resulting in the projected feature map F_{proj}.

Dynamic Position Encoding. Before being processed by the Transformer layers, spatial context is explicitly added to F_{proj}. To achieve this, we employ the identical dynamic center-cropping position encoding mechanism as detailed in Appendix A.1. A positional embedding, P_{crop}, corresponding to the spatial dimensions of the feature map is extracted and added element-wise:

$$\mathrm{F}_{pos} = \mathrm{F}_{proj} + \mathrm{P}_{crop} \tag{4}$$

Transformer Layers. The spatially-aware tensor, F_{pos}, is then fed into a series of standard Transformer layers. The number of layers was a configurable hyperparameter in our experiments, with configurations such as two and six layers being tested, as referenced in the main body of the paper.

The primary hypothesis for this design was that a deeper, more expressive transformation of the VAE features through a dedicated Transformer stack would enable more effective

integration with the broader VLM. However, while functionally sound, our empirical results showed that this approach was not the most efficient in terms of either computational cost or performance gains.

A.3. Overall Architecture and Parameters

A.3.1. Our Architecture Design

To ensure reproducibility and provide a clear reference, we detail the key hyper-parameter settings for the components of our WeMMU model in Table 1. This table covers the specific configurations for all modules, from the Vision Encoder and Large Language Model (LLM) to the Generation Expert and the newly introduced VAE branch. These parameters represent our choices after multiple rounds of experimental optimization, aimed at balancing model performance and computational efficiency. For a more conceptual description of how these components work together within our framework and the design philosophy, please refer to Section 3.2 of the main paper.

Table 1. Hyper-parameter settings for WeMMU models

Module	Param.	WeMMU settings
Vision Encoder(QWen2.5-VL)	N_{layer}	32
	d_{hidden}	1280
	d_{out}	2048
LLM (QWen2.5-VL)	N_{layer}	36
	d_{hidden}	2048
Generation Expert	N_{layer}	36
	d_{hidden}	2048
Linear Layer (after VAE Encoder)	d_{in}	32
	d_{out}	2048
PositionMLP	d_{in}	2048
	d_{inter}	16384
	N_{layer}	1
	d_{out}	2304
VAE (Sana)	d_{in}	3
	d_{hidden}	32
	f_{scale}	0.4107
DiT (Sana)	N_{layer}	20
	$d_{caption}$	2304
	d_{in}	32
	d_{out}	32

A.3.2. Comparative Analysis of Architectural Choices

A central challenge in designing unified multimodal architectures is how to efficiently integrate understanding and generation capabilities. This challenge is particularly acute in scenarios involving a large number of input images, where the comparison between different architectural philosophies becomes most meaningful.

The first paradigm, represented by models like Bagel [10], achieves powerful native image generation capabilities through end-to-end training within a single, unified model. The advantage of this approach lies in its ability to achieve a high degree of instruction following and semantic alignment. However, this comes at the cost of requiring massive, high-quality image-text datasets for training from scratch.

The second paradigm attempts to lower the training cost by "bridging" a pre-trained MLLM with a Diffusion Model, a strategy exemplified by works like MetaQueries [22]. This approach cleverly leverages the powerful capabilities of existing models, allowing the MLLM to focus on understanding complex, multi-source contextual information.

However, to compensate for the loss of fine-grained details during the MLLM's high-level processing, methods like Query-Kontext [25] typically require injecting all the processed contextual features (e.g., dense image features, text embeddings) directly into the Diffusion Model. This is efficient, but as the number of source images increases, a critical bottleneck emerges. In these methods, the volume of contextual information grows dramatically. Injecting this complex and high-dimensional amalgamation of features directly into the Diffusion Model makes fine-tuning it exceptionally difficult. The generator must learn to dynamically disentangle and align features from an arbitrary number of inputs during the diffusion process, a highly unstable and complex optimization task.

In contrast, our WeMMU framework proposes a more advantageous "division of labor" strategy. In our design, the powerful MLLM undertakes the tasks it excels at: deeply understanding and reasoning about complex multimodal inputs, including parsing long text instructions and identifying relationships between multiple images. Crucially, the MLLM does not need to pass all raw or processed features directly to the generative model. Instead, through our proposed Noisy Query Tokens mechanism, it "summarizes" the complex generative intent and encodes it into a concise, robust distributed representation.

Intuitively, this clear division of labor not only preserves the potential for end-to-end optimization but also significantly reduces the "cognitive load" on both core components. We believe this presents a more efficient and promising path toward achieving sustainable and scalable unified multimodal generation frameworks.

A.4. Training Details

Optimizer and Learning Rate. Across all four training stages, we utilized the AdamW optimizer with hyperparameters set to $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay of $1.0e^{-5}$. The learning rate schedule for each stage included a specific number of warmup steps, followed by a cosine decay schedule, as detailed in Section 4.1 of the main paper.

B. Inference Strategy

Our model employs distinct inference-time guidance strategies tailored to the specific task: text-to-image generation,

single-image editing, and multi-image editing. These strategies leverage variations of Classifier-Free Guidance (CFG) to steer the diffusion process. It is important to note that in the following formulations, the noise prediction ϵ is generated by the diffusion model, which is conditioned on the latent representations produced by our Vision-Language Model (VLM) under different input scenarios.

Text-to-Image Generation. For standard text-to-image synthesis, we utilize a conventional CFG approach. Two conditioning latents are generated by the VLM: one from the target text prompt (C_{con}) and another from an empty or null text prompt (C_{uncon}). The diffusion model then predicts the corresponding noises, ϵ_{con} and ϵ_{uncon}. The final noise prediction, ϵ_{pred}, is a linear combination of these two predictions, formulated as:

$$\epsilon_{pred} = \epsilon_{uncon} + \lambda(\epsilon_{con} - \epsilon_{uncon}) \tag{5}$$

Here, λ is the guidance scale hyperparameter that controls the strength of the adherence to the text prompt. A higher value of λ encourages the generation to more closely match the text description. In all text-to-image experiments, the value of λ was set to 4.0.

Single-Image Editing. For single-image editing, a more sophisticated three-component guidance strategy is required to balance fidelity to the source image with the desired textual modification. We compute three distinct noise predictions: (1) ϵ_{uncon}: Predicted from an unconditional latent, generated by the VLM with no image or text input. (2) ϵ_{rec}: Predicted from a reconstruction latent, where the VLM is conditioned on the source image and a simple reconstruction prompt (e.g., "Please reproduce this image"). (3) ϵ_{edit}: Predicted from an editing latent, where the VLM is conditioned on the source image and the target editing prompt (e.g., "a cat wearing a top hat"). The final noise prediction is then calculated by combining these three components. This allows for separate control over reconstruction fidelity and edit strength:

$$\epsilon_{pred} = \epsilon_{uncon} + \lambda_{rec}(\epsilon_{rec} - \epsilon_{uncon}) + \lambda_{edit}(\epsilon_{edit} - \epsilon_{rec}) \tag{6}$$

In this equation, λ_{rec} is the reconstruction guidance scale, which encourages the model to preserve the structure and content of the original image. The term $(\epsilon_{edit} - \epsilon_{rec})$ represents the semantic direction of the edit itself. Consequently, λ_{edit} controls the magnitude of the modification described in the editing prompt. This dual-guidance mechanism is crucial for achieving high-fidelity and semantically coherent image edits. In all Single-Image Editing experiments, the value of λ_{rec} was set to 2.0 and the value of λ_{edit} was set to 3.0.

Multi-Image Editing. When performing edits that involve multiple source images (e.g., style transfer or subject swapping), the concept of a single "reconstruction" target becomes ill-defined. Therefore, we revert to a guidance strategy that is structurally similar to that of text-to-image generation but incorporates multi-image conditioning. Specifically, we compute two noise predictions: (1) ϵ_{uncon}: The standard unconditional noise prediction, as used in the other tasks. (2) ϵ_{multi}: Predicted from a conditional latent, where the VLM is provided with the full set of source images and the target editing prompt. The guidance is then applied as follows:

$$\epsilon_{pred} = \epsilon_{uncon} + \lambda_{multi}(\epsilon_{multi} - \epsilon_{uncon}) \tag{7}$$

Here, λ_{multi} is the guidance scale that determines how strongly the final output should reflect the combined context of the multiple input images and the textual instruction. In all Multi-Image Editing experiments, the value of λ_{multi} was set to 3.0.

C. Additional Generated Images

C.1. Text-to-Image Generation

Fig. 1 showcases a diverse gallery of images synthesized by our 'WeMMU' model. To provide full context for these high-fidelity results, the corresponding text prompts for each image are detailed below, following a top-to-bottom, left-to-right reading order.

1 *The model name 'WEMMU' spelled out with colorful, polished wooden toy blocks on the light wood floor of a child's playroom. The room is flooded with warm, bright sunlight from a large window, creating soft, long shadows. Shallow depth of field.*

2 *A beautiful macro photograph of a slice of pastel rainbow layer cake. On the top, the words 'Sweet Dreams' are written in elegant, creamy white icing, adorned with tiny, colorful sugar sprinkles. The background is a bright, cheerful, out-of-focus party scene.*

3 *A vast, breathtaking crystal cave, where the ceiling is open to the bright sky above. Pure, brilliant sunlight streams down, striking giant, perfectly formed crystals of amethyst and quartz, refracting into a thousand vibrant rainbows that illuminate the entire cavern. A crystal-clear river flows gently through the cave floor.*

4 *A panoramic, sun-drenched vista of a Solarpunk utopian city. Gleaming white towers with organic, flowing lines are covered in lush vertical gardens and waterfalls. Shimmering, elegant glass sky-bridges connect the buildings, and flying, bio-luminescent vehicles drift through the clean air. The scene is bright, optimistic, and incredibly detailed.*

Figure 1. A gallery of diverse text-to-image generation results from our 'WeMMU' model, synthesized at 1024x1024 resolution.

5 *The phrase 'Brewing Happiness' is written in beautiful, elegant white script on the clean glass window of a charming coffee shop. Through the window, the cafe's warm, brightly lit, cozy interior is visible.*

6 *The word 'Elysian' embossed in shimmering gold foil on a pristine, polished white marble wall at the entrance to a luxury boutique. Soft, bright, diffused lighting from above creates gentle highlights and shadows on the letters.*

7 *A single, ancient stone lantern in a serene Japanese garden during a heavy downpour. Its warm, gentle light creates a small haven of peace, and its reflection shimmers beautifully on the wet, mossy stone path.*

8 *A wide-angle landscape shot from a mountain peak. In the foreground, a gnarled, ancient pine tree clings to the rocks. In the mid-ground, a misty valley unfolds. In the far distance, the sun rises behind another range of snow-capped mountains.*

9 *An open, magical book lies on a pedestal in a sunlit library. The pages are made of hammered gold leaf, and as the pages turn, intricate 3D models of trees, animals, and cities emerge from the pages in shimmering, golden light.*

10 *A futuristic, semi-transparent holographic interface displaying glowing blue lines of code and, in the center, the word 'WeMMU' rotates slowly in 3D space.*

11 *A flawless, pure white arctic fox curled up on pristine, sparkling snow under a bright, clear arctic sun. The brilliant light reveals the incredible detail and texture of its thick fur against the crystalline snow. White-on-white photography, every strand visible.*

12 *A whimsical, miniature, exquisitely detailed garden of blooming flowers and tiny, magical creatures, all contained within an elegant porcelain teacup. The scene is shot with a macro lens under bright, soft studio lighting.*

13 *An extreme macro photograph of a Glasswing butterfly (Greta oto) resting on a vibrant, dew-covered orchid. The bright morning light illuminates its stunningly transparent wings, highlighting the delicate, iridescent borders. Every tiny detail is in perfect focus.*

14 *A beautiful clockwork hummingbird, crafted from polished gold, gleaming silver, and tiny, sparkling sapphires. It hovers beside a vibrant, sun-drenched hibiscus flower, its intricate gears and mechanisms fully visible and crisply rendered. Macro photography, bright natural light.*

15 *A majestic whale gracefully swimming through the clouds, its body a stunning fusion of organic flesh and intricate, polished brass clockwork mechanisms. Steampunk, fantasy.*

16 *A serene and beautiful woman with delicate features, standing in an orchard of blooming white apple blossoms. Bright, soft sunlight filters through the branches, highlighting the translucent quality of the petals and the soft texture of her skin.*

17 *Ultra-realistic portrait of a beautiful young woman with a gentle smile, sitting inside a cozy, brightly lit cafe. Soft natural light streams through a large window covered in fresh raindrops. The warm interior glow reflects in her eyes, and you can see the delicate texture of her skin and the steam rising from her coffee cup.*

18 *A majestic griffin perched on a sun-drenched marble cliff. Its magnificent wings are not made of feathers, but of intricate, interlocking pieces of opalescent crystal that refract the bright sunlight into a dazzling spectacle of color.*

19 *A majestic and benevolent forest spirit, resembling a great stag, whose body appears to be woven from pure, solid light and living, green leaves. Its grand antlers glow with a warm, gentle light. It stands peacefully in the center of a sun-dappled, magical forest, radiating an aura of calm and life.*

20 *A vibrant, emotional anime scene in the style of Makoto Shinkai. A beautiful young woman with sparkling eyes stands under a blooming cherry blossom tree as a train passes by. The wind sweeps petals around her. The scene is characterized by dramatic lens flare, incredibly detailed backgrounds, and a bright, saturated color palette.*

21 *A beautiful, gentle young woman with a warm smile, in the charming, hand-drawn aesthetic of Studio Ghibli. She is sitting in a lush, sunlit meadow, weaving a crown from a basket of fresh wildflowers. The scene is peaceful, nostalgic, and filled with warm, natural light.*

C.2. Single-Image Editing

Beyond image generation, our 'WeMMU' model also supports a variety of single-image editing tasks. Fig. 2 showcases this versatility across a range of creative and functional edits, with prompts annotated directly on the images. The demonstrated capabilities include fundamental semantic changes (adding, removing, replacing), stylistic transfers (origami, sketch), and utility-oriented functions such as depth map generation and virtual try-on, highlighting the model's ability to respond to a diverse set of user instructions.

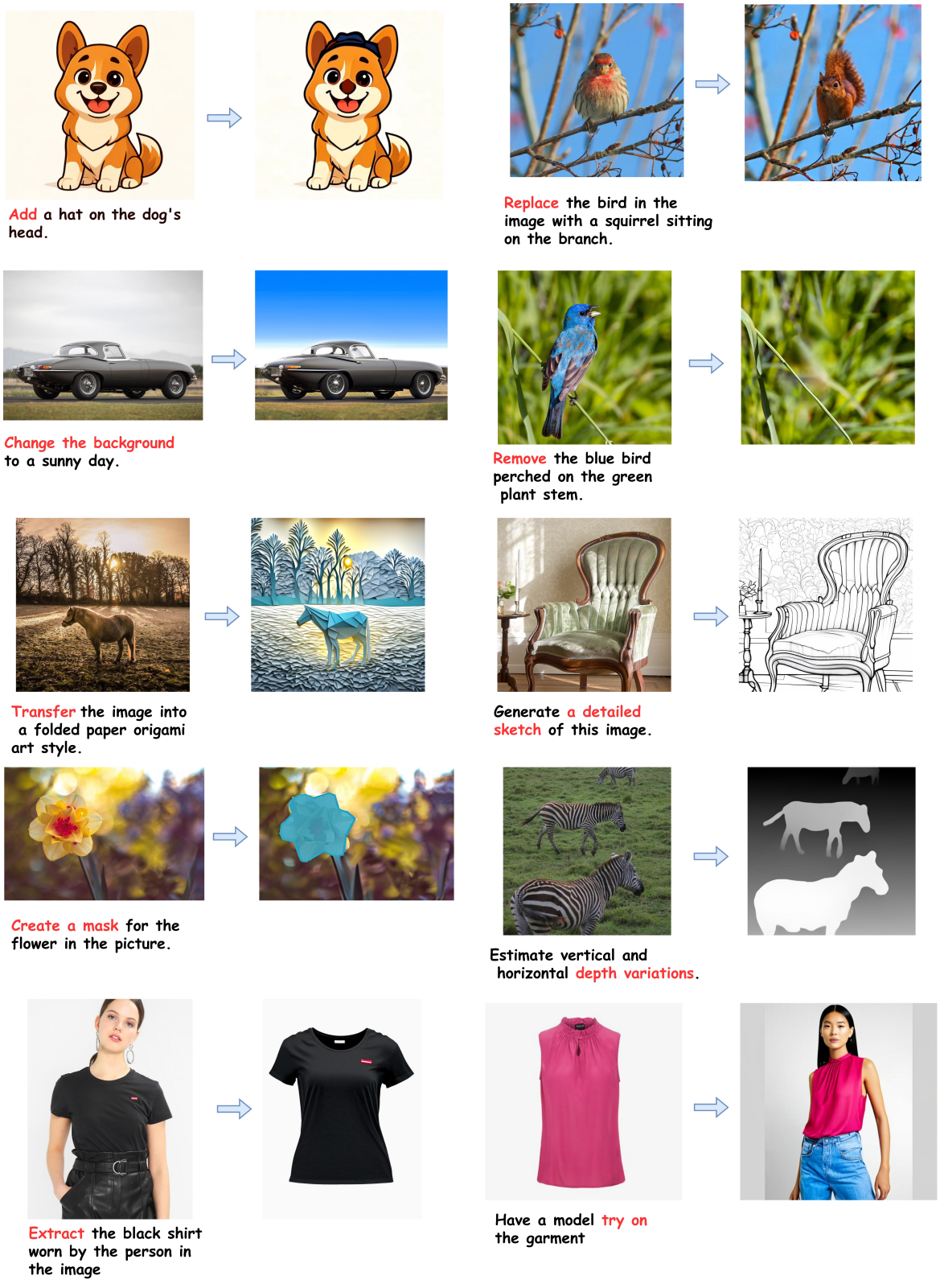

Figure 2. Illustrative examples of single-image editing performed by our 'WeMMU' model. Each pair shows the original image (left) and the edited result (right) based on the annotated prompt.

Figure 3. Illustrative examples of multi-image editing. Each set displays the two source images alongside the final composite image generated by 'WeMMU' based on the annotated prompt.

C.3. Multi-Image Editing

Finally, we explore the model's capabilities in multi-image editing, a task that requires synthesizing a new image based on two source images and a guiding text prompt. Fig. 3 presents examples of this functionality. While the results demonstrate that 'WeMMU' is capable of following these compositional instructions, successfully identifying, extracting, and arranging specified features from both source images into a new target image, we also observe certain artifacts in the final outputs. Some generated images may exhibit a noticeable "pasted-on" appearance at the seams of integrated elements, occasionally accompanied by a loss of fine-grained detail. We attribute these limitations primarily to the scarcity and inconsistent quality of the multi-image conditioning data used during training.

Nevertheless, these examples validate that the core mechanism for multi-image feature extraction and compositional synthesis is functional. This establishes a promising foundation for future work, where performance can be substantially improved with access to larger, higher-quality datasets.

D. Additional Analysis on Training Efficiency

To provide a comprehensive understanding of WeMMU's performance in the context of computational cost, we compare its training efficiency with recent unified models, such as Bagel [10] and OmniGen2. Model performance is inherently coupled with model size, data scale, data quality, and algorithmic design. While WeMMU utilizes 25.8M public data samples, a smaller scale compared to the proprietary or larger-scale datasets used by OmniGen2 (151M) and Bagel (1600M), it achieves highly competitive performance with significantly lower computational overhead.

As shown in Table 2, WeMMU achieves comparable generation and editing capabilities with approximately $1/10$ of the training FLOPs required by OmniGen2. We calculate the training FLOPs as $6 \times N \times D$, where N is the total number of trainable parameters and D is the total number of training image tokens. Since OmniGen2 and Bagel employ mixed-task training, we conservatively estimated their FLOPs using their minimum reported ratios. This high training efficiency validates the effectiveness of our lightweight bridging design.

Table 2. **Training Efficiency Comparison.** FLOPs are estimated based on total trainable parameters and training tokens.

Methods	Dataset Size	Model Size	Training FLOPs
WeMMU	25.8M	8B	1.66×10^{21}
Bagel	1600M	14B	$> 1.57 \times 10^{23}$
OmniGen2	151M	7B	$> 2.10 \times 10^{22}$

E. Continual Learning and Forgetting Analysis

A critical challenge in training unified multimodal models across progressive stages is catastrophic forgetting, where the model loses proficiency in earlier tasks (e.g., text-to-image) while learning new ones (e.g., image editing). To explicitly evaluate WeMMU's sustainability, we quantify the Forgetting Measure (FM) as the performance drop on old tasks after learning new tasks. Specifically, the FM between Stage 3 (learning single-image editing) and Stage 4 (learning multi-image editing) is defined as $FM = P_{S3} - P_{S4}$.

Table 3. **Continual Learning Analysis.** Forgetting Measure (FM) between Stage 3 and Stage 4. Lower values indicate less forgetting, and negative values indicate positive transfer.

Benchmark	Metric	Stage 3	Stage 4	FM (↓)
GenEval (Gen.)	Overall	0.88	0.88	**0.00**
DPG-Bench (Gen.)	Overall	83.69	83.60	**0.09**
ImageEdit (Edit)	Overall	3.31	3.30	**0.01**
GEdit-Bench (Edit)	Overall	5.75	5.77	**-0.02**

As presented in Table 3, our model achieves near-zero forgetting and even exhibits positive transfer (a negative FM of -0.02) on GEdit-Bench. This robustness is attributed to two key mechanisms:

1. **Data Replay:** We maintain a carefully tuned mixture of previous task data during new training stages to serve as functional anchors.
2. **Plasticity via Noisy Query Tokens:** Replacing static learnable parameters with Noisy Query Tokens provides a low-overhead solution to prevent overfitting in the latent bridge. From an attention perspective, this shifts the model's focus from dominating image tokens to text instruction tokens, preventing the "task generalization collapse" (i.e., the solidification of learnable queries causing a loss of plasticity).

Furthermore, WeMMU rapidly acquires basic multi-image editing capabilities in Stage 4 using only a limited amount of task-specific data. We also explicitly tested zero-shot inference on unseen multi-image editing tasks without any fine-tuning. The model failed to correctly fuse the images, confirming that while our architecture provides a highly plastic foundation, task-specific data remains essential for activating novel capabilities.

F. Extended Ablation Studies

Learnable Fixed Query vs. Noisy Query Tokens. In our main ablation study (Section 4.3 of the main paper), the "Learnable Fixed Query" baseline represents an adapted version of the widely-used MetaQueries approach. Under strictly controlled settings, switching from the Learnable Fixed Query to our proposed Noisy Query Tokens significantly improves the ImageEdit score from 2.53 to 3.31.

This quantitatively validates that our approach overcomes the limitations of the standard learnable token paradigm.

Robustness Across Different Architectures. To further demonstrate that the effectiveness of Noisy Query Tokens is not limited to a specific network architecture, we extended our evaluation to two distinct setups: (1) a smaller MLLM backbone (**LLaVA-OV-0.5B** paired with Sana1.5-1.6B) and (2) a different diffusion backbone (Qwen2.5-VL paired with **SD3.5-Medium**). To facilitate efficient experimentation, these models were trained on a reduced dataset (approximately 10% of the full schedule), ensuring strict comparability between the baseline and our method.

Table 4. **Ablation on Different Backbones (Reduced Data).** Evaluation of task generalization collapse across various architectures.

Setup	Method	ImageEdit	Task Collapse?
LLaVA-OV	Learnable Query	2.13	Yes
+ Sana1.5	**Ours**	**2.69**	**No**
Qwen2.5	Learnable Query	2.33	Yes
+ SD3.5	**Ours**	**2.91**	**No**

As shown in Table 4, the *Learnable Query* baseline consistently reverts to basic image reconstruction (i.e., suffering from Task Generalization Collapse) across diverse architectures. In contrast, our *Noisy Query* approach consistently maintains instruction adherence and yields superior editing scores. This confirms that our mechanism is robust, model-agnostic, and fundamental to the architecture's success.