BULaMU-Dream: The First Text-to-Image Model Trained from Scratch for an African Language

Mwebaza Rick
ricky.mwebaza@gmail.com

Abstract

This paper introduces BULaMU Dream, the first text-to-image, conditional diffusion model in the world trained from scratch to generate images from prompts in an African Language, Luganda.

Introduction

Multimodal learning has been around for several years now, with technologies like OpenAI's Sora and Google's Nano Banana improving at a breakneck pace (Cai et al., 2025). While multimodal large language models are trained on expansive, internet scale datasets, they often struggle with prompts in African Languages, due to the lack of large, high quality training corpora (Welle, 2025). This paper introduces BULaMU Dream, the first text to image diffusion model in the world trained from scratch to generate images from prompts in an African Language, Luganda.

BULaMU-Dream

BULaMU-Dream employs Google's Imagen architecture, consisting of a text encoder (T5) that converts Luganda text into a sequence of embeddings that are mapped by an ensemble of machine learning models to 28x28 images. BULaMU Dream was trained on a modified version of the Fashion-MNIST, which includes 60,000 28x28 grayscale images of fashion products from 10 categories in its "train" set. These 10 category names from the original dataset were translated to Luganda and are as follows: "Bbuutu y'enkizi", "Ekiteeteeyi/eky'okungulu","Ekiteteeyi", "Empale", "Ensawo", "Kooti", "Pulover ya Pulover", "Saati", "Sandal", and "Sneaker". The following subsections detail the different components of the model architecture and the training process.

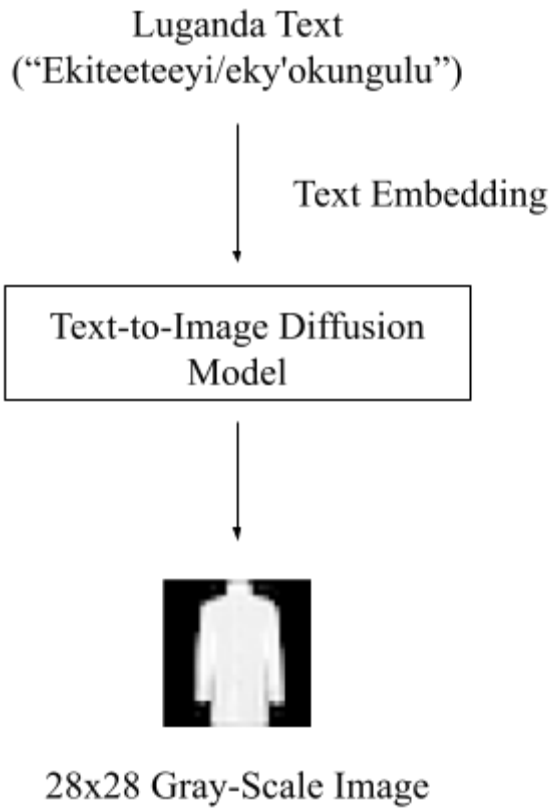

Figure 1: Illustrates the architecture of BULaMU-Dream

Encoding Luganda Text

Initially, I planned to prune Google’s pretrained mT5-base model (580M parameters) using the same custom 390-million-token dataset that I used to train BULaMU to create a text encoder specifically for Luganda (Mwebaza, 2025). The idea was to eliminate unnecessary multilingual capacity and focus the model’s representational power on a single low-resource language. However, this approach proved ineffective since the pruned model struggled to distinguish between classes, which was reflected in consistently high cosine similarity across unrelated labels. mT5 relies on sentencepiece subword tokenization and maybe the amount of Luganda in the sentencepiece training corpus was prohibitively small. As a result, many Luganda words are mapped to weak or overlapping subword representations. When the model was pruned, this limitation became even more pronounced, making it difficult to preserve meaningful semantic distinctions between the different classes in my dataset.

I switched to ByT5 because it is language agnostic. It encodes input text strings as UTF-8 and directly maps the resulting bytes to embeddings, which makes it well suited for low-resource languages like Luganda since it does not rely on a fixed subword vocabulary. Unlike mT5, ByT5 was able to distinguish between classes effectively, as shown by the lower cosine similarity between embeddings from different labels. This indicated that the model was learning

meaningful semantic differences rather than collapsing multiple classes into similar representations.

Training U-Net Model

I trained the diffusion UNet conditioned on ByT5 embeddings for Luganda labels for 20,000 steps on a base M4 Mac Mini, which has a 10 core CPU, 10 core GPU, and 16GB of RAM.. The ByT5 encoder produces 1,472-dimensional embeddings, and I configured a lightweight UNet for 28×28 Fashion-MNIST images stored as RGB. The UNet used a base width of 64 channels with dimension multipliers (1, 2, 4) and two ResNet blocks per stage. For efficiency on Apple Silicon (MPS), self-attention was enabled only at the coarsest stage (False, False, True), while cross-attention to the text embeddings was enabled at the latter two stages (False, True, True).

To speed up training, I precomputed and cached text conditioning on-device: since there were 10 unique captions/labels, I embedded each caption once with ByT5 and stored the resulting embedding table directly on MPS with shape. During training, batches simply indexed into this table instead of re-encoding text every step. I trained with batch size 256, learning rate 1e-4, and 250 diffusion timesteps. Training ran at roughly 0.66–0.69 steps/sec, with loss dropping rapidly early (0.57 at step 25 → ~0.13 at step 100) and stabilizing around ~0.015–0.02 later. I saved checkpoints every 2,500 steps.

Results

Figure 2: Displays a sample image from each class in the Fashion-MNIST dataset and 3 images generated by BULaMU-Dream for each prompt.

I wrote a Python script to generate three images for each prompt, which can be seen in the figure above. After training for 20,000 steps, BULaMU-Dream was able to consistently generate high quality images for each class. The model would occasionally generate images that didn't correspond to the correct class, which can be seen above where the model generates pants ("empale") when it was prompted to generate a coat ("kooti"). This issue can most likely be rectified by allowing the U-Net to train for more steps.

Discussion

This paper introduces BULaMU-Dream, the first conditional diffusion model in the world trained from scratch to generate images from text prompts in Luganda, an African Language. This model shows that given enough high quality, labeled data, which can often be adapted from corpora in "high-resource" languages like the Fashion-MNIST dataset, artificial intelligence and machine learning models can understand semantic differences between words in "low-resource", African Languages like Luganda given enough examples to learn from and respond accordingly. BULaMU-Dream also shows that it is possible not only to inference but also train conditional diffusion models from scratch for low resource languages using relatively inexpensive setups, such as the base M4 Mac Mini.

BULaMU-Dream would greatly benefit from a few improvements. Even though the model is able to consistently generate images that correspond to the prompts it is given, it occasionally outputs images that don't correspond to the correct class, as seen in Figure 2, where it generates pants ("empale") when it was prompted to generate a coat ("kooti"). This can most likely be rectified by allowing the model to train for more steps. As mentioned earlier, I also used Metal Performance Shaders (MPS) to accelerate the training of this model in PyTorch, which would often cause memory leakage and high swap memory usage on my M4 Mac Mini (rbosh, 2025).

References

Cai, Z., Qiu, H., Ma, T., Zhao, H., Zhou, G., Huang, K.-H., Kordjamshidi, P., Zhang, M., Wen, X., Gu, J., Peng, N., & Hu, J. (2025). MMGR: Multi-Modal Generative Reasoning. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2512.14691

lucidrains. (2025). *lucidrains/imagen-pytorch: Implementation of Imagen, Google's Text-to-Image Neural Network, in Pytorch*. GitHub. https://github.com/lucidrains/imagen-pytorch

Mwebaza, R. (2025). BULaMU: An Open Foundation Model for Luganda. *Zenodo (CERN European Organization for Nuclear Research)*. https://doi.org/10.5281/zenodo.17271688

rbosh. (2025, June 3). *MPS Memory Leak*. GitHub. https://github.com/pytorch/pytorch/issues/155060

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Kamyar, S., Ghasemipour, S., Karagol, B., Mahdavi, S., Lopes, R., Salimans, T., Ho, J., Fleet, D., & Norouzi, M. (n.d.). *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. https://imagen.research.google/paper.pdf

Welle, D. (2025, September 2). *Mind the gap: Building inclusive AI for African languages*. Deutsche Welle. https://akademie.dw.com/en/mind-the-gap-building-inclusive-ai-for-african-languages/a-73693134