# VisualFusion: Enhancing Blog Content with Advanced Infographic Pipeline —Supplemantary Material—

Anurag Deo*
Indian Institute of Technology Patna
anurag_2101ai04@iitp.ac.in

Savita Bhat
TCS Research
savita.bhat@tcs.com

Shirish Karande
TCS Research
shirish.karande@tcs.com

## 1. Prompts

In the complete pipeline we have prompted different LLMs to generate responses for codes and stable diffusion prompts. In this section we are going to pen down the prompts that we have used to generate the codes and prompts.

### 1.1. Prompt for Code Generation

The prompt for the code generation is as follows:

```
You are an exceptionally intelligent
coding assistant that consistently
delivers accurate and reliable
responses to user instructions.
Instructions:  Instructions:
You are given a table in the format of
a 2D list with some data in it.
The value are in string format so while
writing in the code make sure you add
them as numbers for example value like
'50%' should be written just 50 in
the code so that there will be fair
numerical comparision.
You need to plot only ONE COLUMN of
the table in the graph which you think
is most suitable and important for
example if there is some column which
represent the share out of total.  Add
that chosen column to the response as
'''column then the column name '''.
You need to write a python code to
generate bar graph.
Make double sure that THE RESPONSE
SHOULD ONLY CONTAIN THE CODE AND COLUMN
NAME IN MARKDOWN format.
Make double sure to DON'T ADD LEGENDS,
AXES AND LABELS TO THE GRAPH, If you
are using matplotlib for bar graph then
you can do this using plt.grid(False)
and plt.axis('off').
Make sure to import the necessary
libraries create the dataframe from the
text and at the end save the image as
img_name.png Make no assumptions while
writting the code as the code will be
run on the server side.
Using the table provided below you
have to create the table variable which
stores the table which you want to use
for the graph and then write the code
to generate the graph.  You are not
allowed to take it from any file or any
other source.
STEPS TO FOLLOW:


 1. Create a table variable and store
    the given table in it.

 2. Think and reason which column is
    most suitable for the bar graph.

 3. Do some basic data cleaning in 'that
    column only' like removing the '%',
    '$' or any other sign from the
    values.  Removing the commas from
    the numbers.

 4. DO ONLY FOR SELECTED COLUMN: If
    the selected column is something
    related to money then it will have
    letter like K denoting thousands,
    M denoting millions, B denoting
    billions etc.  Remove these letters
    from that column only and then
    convert the values to float or int
    and then multiply the values with
    1000, 1000000, 1000000000 depending
```

*This work was conducted while the author was an intern at TCS Research.

on the alphabet removed. Do similar thing if you encounters words like million, billion, giga, etc

5. DO ONLY FOR SELECTED COLUMN: Convert the values inside SELECTED column to float or int so that you can plot them easily.

6. Write the code to generate the bar graph for that column only.

Response Format:
```python
Code for the plot generation will be here
```
```column
column name
```
Table

### 1.2. Prompt for SD Prompt Generation

For generating the SD prompt we have followed a chain of thought (COT) mechanism of prompting. The complete prompt for that is as follows:

I have a blog for which i need to generate an aesthetic infographics which will draw user's attention and increase the viewership. For that i have generated a simple bar graph for my data using matplotlib and now i want to use stable diffusion to beautify my graph and your task is to provide me a prompt less than 77 tokens which i will give to te stable diffusion to implaint the bars of the bar graph. The steps to generate a good prompt are listed below:

1. First according to the text given below find out the most appropriate item that can be incorporated in the bars of the bar graph. The item that needs to be selected must align with the theme of the text and also should fit in the bars of the bar graph. Try to avoid any object whose width is much larger than height and depth. The selected item will be added in the bars of the bargraph.

2. After selecting the item give proper description about how the item will be used as bars of the graph like standing upright front view or stacked on each other etc, its look, color, whether it is shiny or dull, of which material it is built any other description like something flowing out of it, something put inside or outside it. If you are unable to give proper description try to think on some other related item for which you can think of the description.

3. Give description about the background in which the items you mentioned above should be kept, give proper response including how the object will interact with the background for example will it be over the surface, of on the ground, or submerged under the water, or floating on water etc. The theme of the background you give must aligns to the text given below

4. Give the lighting conditions whether it is well lit or dim or sunny or night. Whether there are some reflection or not etc.

5. Next add the word ultra-realistic, 8k, digital art, focus, sharp at the end.

6. Summarize everything that you get till now so that the final prompt is around 77 tokens

@@TEXT:
Text from blog will go here
With small changes this prompt can be adapted for generating pie charts and area plots.

### 1.3. Prompt for Caption Generation

The caption is generated using the following prompt:

You are an excellent data scientist who has been given a table in the format of a 2D list with some data in it. The values are in string format so while writing in the code make sure you add them as numbers. For example, a value like '50%' should be written as just 50

```
in the code so that there will be fair
numerical comparison.
You are also given a column name, now
i have the plot for that column name.
Your task is to generate a good 4-5
words title for the graph.  It should
align with the data for example it
can be 'increasing prices of oil' or
'Increasing pollution due to plastic'
etc.
Now you need to give a description for
the image of the plot which i will use
in the blog.  For more reference i am
giving you the blog also which i have
written.  The description will be such
that it should give the insights from
the table about the column name and
should be a good 2-3 lines long.  For
example if the column name is 'oil
prices' and the trend that is seen
in the table is incresing till 2010
and highest in 2020, then the caption
can be 'The prices of oil have been
increasing since 2010 and have reached
the highest in 2020'.
For this task you can follow the
following steps:


 1. First take out the data for that
    column and try to find out what the
    data is all about.

 2. Then try to find out the statistics
    of the data like its mean,median,
    mode, maximum, minimum etc whatever
    you think is necessary.

 3. Then try to find out the trend in
    the data like is it increasing,
    decreasing or constant.

 4. Then try to find out the insights
    from the data and text like what can
    be the reason for the trend, what
    can be the future trend etc.

 5. Then try to write a good title and
    description for the image of the
    plot.

The table is as follows:
{table}
The column name for which you have to
generate the title and description is:
{column}
The blog content is as follows:
{blog}
Response should be like this no
extra spaces or lines in between
the response.  Just the title and
description in the following format:
'''
Title
'''
'''
Description
'''
```

## 2. Human Evaluation Guidelines

For the human evaluation, we selected five undergraduate interns from our organization, each with a background relevant to the task. To ensure consistency and reliability in their evaluations, each evaluator was provided with 10 unique images and 20 images also evaluated by other participants.

Prior to the evaluation, the evaluators attended a training session where they were briefed on the evaluation criteria and process. They received clear instructions: they were presented with an anchor plot representing the actual data, along with the title and summary of the blog. Their task was to read the blog summary, examine the anchor plot, and then evaluate three images generated by our pipeline. They were instructed to select the image that best aligned with the blog's content and data representation, based on criteria such as accuracy, relevance, aesthetics, and data adherence.

Each data instance received votes from three evaluators. The image with the majority vote was considered the best for that instance. In cases where each evaluator selected a different image, resulting in a tie, a master evaluator—an experienced graphic designer—reviewed the images and made the final selection.

The evaluators' votes were collected and recorded systematically to ensure transparency. Measures were taken to minimize biases, such as randomizing the order of images shown to evaluators and ensuring they were unaware of which image was actually the best among them.

After this process, we identified the best image for each data instance. These selected images were then compared to the images identified as best according to the AADaT score. We calculated the number of matches between the human-selected best images and those determined by the AADaT score to assess the alignment of our evaluation metric with human perception.

## 3. Hyperparameters used in the pipeline

The table below shows the value of hyper parameters used in the pipeline

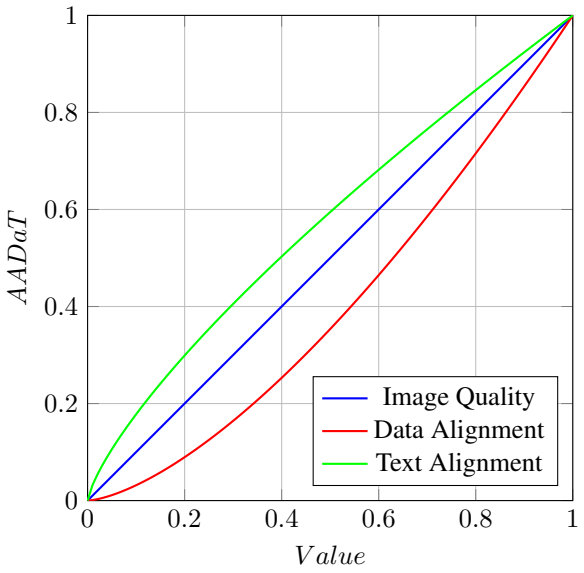


Figure 1. Variation of AADaT score based on different parameters

| Hyper parameter | Value |
|---|---|
| Temprature | 0.4 |
| Controlnet's max-sequence-length | 100 |
| Canny Low Threshold | 100 |
| Canny High Threshold | 200 |
| Controlnet conditioning scale | 0.5 |

Table 1. Table showing the comparison between the Llama-70B and GPT-4o

## 4. Infographics Analysis

Infographics created by professional graphic designers often exhibit a level of complexity that surpasses our current capabilities. Our model represents a step towards achieving comparable results in the field.

As shown in figure 2 the infographics [1] broadly consists:

- A primary plot, such as a bar graph, pie chart, or line chart,
- Surrounding text designed to enhance interpretability,
- A well-chosen background to enhance visual appeal.

Our approach initially focuses on generating plots, with foreground and background elements selected based on textual content from associated blogs. However, enhancing the textual components poses a significant challenge. Effectively integrating meaningful insights extracted from textual sources into the infographic image requires careful consideration of layout, typography, and positioning. This complexity arises from the need of multiple text instances and the need to optimize their visual presentation within the image.

Future directions for our research include addressing these challenges to produce infographics that rival those crafted by experienced designers. Meanwhile, our generated images offer a practical solution for designers seeking aligned visual representations of textual content.

This ongoing work serves to bridge the gap between automated infographic generation and the sophisticated designs produced by professionals, providing valuable tools for various applications in visual communication.

[1] All the infographics are taken from https://www.visualcapitalist.com/

## 5. Analysis of the generated anchor plots

The anchor plot generated by executing the python script generated using LLM can contain potential errors. The primary errors among them are:

- Error in code execution
- Wrong values for the plot generation

### 5.1. Error in code execution

Since the code is generated using language models (LLMs), there is a possibility of encountering various issues, such as invalid syntax, incomplete code, incorrect data types, or other errors. These issues pose significant challenges as the model expects to generate a plot, but these errors can prevent the successful creation of the plot.

To address this problem, we propose incorporating a more robust model into the pipeline, which is capable of correcting the mistakes made by the smaller model, thereby enhancing the overall reliability of the process. Despite this enhancement, a small number of errors may still persist, which can be manually resolved.

Our studies indicate that when relying solely on an 8-billion parameter LLM, the accuracy of generating error-free code is approximately **83%**. However, by introducing a more advanced model, such as GPT-4o, to correct the errors in the generated code, this accuracy increases significantly to **94%**.

### 5.2. Wrong values for the plot generation

Inaccuracies in the generated plots can largely be attributed to the incorrect parsing of data from the table by the language model. The dataset encompasses a diverse range of domains, including finance and green energy, resulting in a table that contains varied data types. A significant issue arises because not all the data is numeric; some values are expressed in formats such as millions and billions, while others include units like kilowatt and megawatt. The language model occasionally fails to convert these values into actual numeric forms before generating the plot, leading to substantial inaccuracies.


Figure 2. Infographics designed by professional graphics designers

To verify the correctness of the numerical values used in the generated code, we employed the LLama 70B model in inference mode. This model takes the code and the corresponding table as inputs and ensures that the code correctly uses the table with accurate numeric representations, either as integers or floats. The prompt used for this task is as follows:

```
You are given a code to generate
a plot written in Python using the
Matplotlib library. Additionally,
you are provided with a table. The
same table is used in the code to
generate the plot. However, there
may be discrepancies between the
provided table and the table used
in the code. The table in the code
has undergone transformations, such
as converting units to their numeric
forms (e.g., millions replaced by
multiplying the value by 1,000,000).
Similar conversions are applied to
other unit prefixes like kilo and
mega. Your task is to verify whether
the table in the code aligns with the
provided table. You need to check
each cell individually to identify any
discrepancies. The input will be a
code and a table in string format, and
the desired output is either "CORRECT"
or "INCORRECT". You are required to
answer in only one word.
```

To address the underlying issue, we have enhanced the prompt to identify units and multiply the corresponding numeric values by the appropriate factors, ensuring all data is converted into a consistent numeric format. Additionally, we implemented a small script within the code to perform similar conversions. However, since this script operates as conditional code, it can only handle a limited number of cases. The remaining values are corrected using the language model, thereby improving the accuracy of the generated plots.

### 5.3. Mitigation of the inconsistencies caused by the stochastic nature of diffusion models

Fine-tuning pre-trained diffusion models for controlled image generation while maintaining diversity is an active research area. Papers like [1, 2] demonstrate that either entropy regularization during fine-tuning can mitigate the

inherent stochasticity of these models or we can train the diffusion model by incorporating a loss that minimizes the sampling drift.

The core idea is to define a reward function $r(Y)$ capturing desired image properties like aesthetics, quality, and data adherence. Fine-tuning then seeks a distribution $\mathrm{P}_{\text{tune}}(\cdot)$ maximizing the expected reward while controlling divergence from the pre-trained model's distribution $\mathrm{P}_{\text{pre}}(\cdot)$.

This can be achieved using entropy regularization:

$$\mathrm{P}_{\text{tune}}(\cdot) = \underset{\mathrm{P}(\cdot)}{\operatorname{argmax}}\, \mathbb{E}_{\mathrm{P}(\cdot)}[r(Y)] - \alpha \mathrm{D}_{\mathrm{KL}}(\mathrm{P}(\cdot), \mathrm{P}_{\text{pre}}(\cdot)),$$

or the more general f-divergence regularization. As in our case we need to optimize the image quality, aesthetics, data adherence and text alignment all at the same time so in our case it will become a multi objective optimization problem which is difficult to model and implement. But we are actively working to mitigate this stochastic nature of the diffusion model.

A key challenge is the lack of sufficient datasets for fine-tuning with complex reward functions. We are working on development of large-scale datasets specifically tailored for this task.

# 6. Intuition Behind the AADaT Metric

The AADaT (**A**esthetics and **A**dherence to **Da**ta and **T**ext) metric was developed to address the limitations of existing evaluation metrics when applied to generated images, particularly those that must align closely with textual descriptions and data. Traditional metrics such as SSIM, IS, FID, and CLIP Score evaluate specific aspects of generated images but fall short of simultaneously assessing text alignment, data adherence, image quality, and aesthetics. The AADaT metric is designed to fill this gap by providing a comprehensive evaluation that balances these four critical components.

## 6.1. Key Components of AADaT Metric

The AADaT metric is defined by the equation:

$$\eta = \chi S_{IQ} S_D \sqrt{1 - |\chi - S_D|} \tag{1}$$

where $\chi = \sqrt{S_A S_T}$, and $S_{IQ}$, $S_D$, $S_A$, and $S_T$ are the image quality score, data alignment score, aesthetics score, and text alignment score, respectively.

### 6.1.1 Penalizing Bias Between Data Alignment and Text Alignment

The term $\sqrt{1 - |\chi - S_D|}$ serves as a penalty function that discourages bias towards either text alignment or data alignment:

- **Penalty for Imbalance:** If the generated image is too biased towards text alignment ($\chi$ is much greater than $S_D$) or data alignment ($S_D$ is much greater than $\chi$), the penalty term $\sqrt{1 - |\chi - S_D|}$ becomes small. This reduction reflects a lower overall score, indicating that the image is unbalanced in its adherence to both text and data.
- **Encouraging Balance:** The penalty term ensures that the evaluation metric favors images that strike a balance between text and data alignment. When $\chi$ and $S_D$ are close, implying that the image adheres well to both the textual description and the data, the penalty is minimized, and the overall score remains high.

### 6.1.2 Direct Dependence on Image Quality, Data Adherence, and Overall Alignment

The final evaluation metric, $\eta$, is directly proportional to the image quality score ($S_{IQ}$), data alignment score ($S_D$), and the balanced alignment score $\chi$:

- **Image Quality ($S_{IQ}$):** High-quality images are essential for ensuring that the generated visuals are not only accurate but also visually appealing. By incorporating $S_{IQ}$, the metric rewards images that are free from distortions and other visual artifacts.
- **Data Adherence ($S_D$):** Since the generated image must accurately represent the underlying data, $S_D$ plays a crucial role in the metric. This score ensures that the image reflects the data correctly, which is especially important in domains where data fidelity is critical.
- **Unified Metric ($\chi$):** The combination of $S_A$ and $S_T$ into $\chi$ ensures that the image aligns with the text and maintains aesthetic quality, providing a holistic evaluation of how well the image conveys the intended message.

# References


[1] Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *Advances in Neural Information Processing Systems*, 36, 2024. 5

[2] Wenpin Tang. Fine-tuning of diffusion models via stochastic control: entropy regularization and beyond. *arXiv preprint arXiv:2403.06279*, 2024. 5

Figure 3. The top two images in the figure exhibit strong data adherence but fall short in terms of aesthetics. The middle two images demonstrate high aesthetics, yet they compromise on data adherence. The final two images present a balanced outcome, effectively showcasing both data adherence and aesthetics.