Progressive Compositionality In T2I Generative Models

Ambati Rahul Reddy - B22CS088 **Paper Link**

PART A

1Q)

1. Problem & Motivation

- The paper addresses a key failure point in even state-of-the-art text-to-image (T2I) diffusion models: their inability to correctly handle compositional relationships.
- This results in common errors like incorrect attribute binding (e.g., swapping colors between objects), miscounting objects, and misunderstanding spatial relationships. For instance, a prompt for "a red motorcycle and a yellow door" might incorrectly produce an image of a yellow motorcycle.
- As T2I models become more powerful and widely used, this lack of compositional understanding limits their reliability for complex, real-world applications. Existing solutions, which often modify model architecture or assume fixed compositional structures, lack the flexibility to generalize well.

2. Core Idea

The core idea is **EVOGEN**, a framework that teaches diffusion models compositionality through a progressive curriculum combined with contrastive learning. Instead of tackling complex scenes all at once, the model learns progressively, starting with simple concepts and building up to more difficult ones.

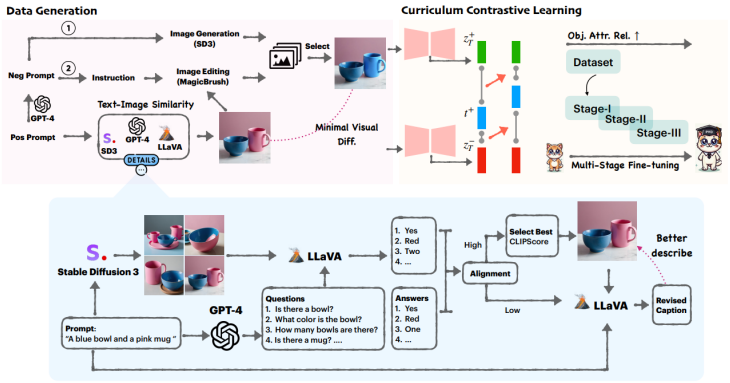


Figure 1: Data Generation (Left) and Curriculum Contrastive Learning (Right).

Data Generation (Left): An automated pipeline uses LLMs (GPT-4) to create positive and negative text prompts. Diffusion models generate candidate images, which are then filtered by a Visual-Question Answering (VQA) model that decomposes the prompt into questions to ensure the final image faithfully matches the text. This process creates high-quality, contrastive image pairs with minimal visual differences.

Curriculum Contrastive Learning (Right): The model is fine-tuned in a multi-stage curriculum, moving from simple to complex tasks to build a foundational understanding.

Key Design Decisions & Equations:

- **Three-Stage Curriculum:** The training is broken into three stages: (1) single object-attribute composition, (2) attribute binding between two objects, and (3) handling complex scenes with multiple objects and relationships.
- **CONPAIR Dataset:** The creation of a new dataset, CONPAIR, with 15k high-quality contrastive image pairs is central to the method.
- **Minimal Visual Differences:** The contrastive image pairs are designed to be minimally different, forcing the model to focus on the specific compositional error (e.g., a swapped color) rather than other visual changes.

InfoNCE Contrastive Loss: The model is trained using a contrastive loss function that pushes the model to maximize the similarity between a text prompt and its correct (positive) image, while minimizing the similarity with its incorrect (negative) image. The loss for a positive-negative pair is defined as:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\text{sim}(f(t), h^+)/\tau)}{\exp(\text{sim}(f(t), h^+)/\tau) + \exp(\text{sim}(f(t), h^-)/\tau)}$$

where $f(t)$ is the text representation, h^+ and h^- are positive and negative image representations, and τ is a temperature parameter.

3. Main Contributions

The paper's main technical contributions are:

- **CONPAIR Dataset:** The introduction of a new, large-scale dataset of 15k contrastive image pairs specifically designed for learning compositionality. A key feature is its automated generation pipeline using LLMs for prompt creation and VQA models for quality control and alignment checking.
- **EVOGEN Framework:** A novel, multi-stage curriculum for contrastive fine-tuning of diffusion models. This progressive learning strategy enables the model to build a foundational understanding of simple compositions before advancing to more complex scenes, significantly boosting performance.

4. Takeaway

The main takeaway is that a structured, progressive curriculum is a highly effective method for teaching abstract concepts like compositionality to large generative models . Simply training on a mixed bag of difficult data can overwhelm the model. By starting with simple tasks and gradually increasing complexity, the model can build a robust foundation that allows it to generalize better to intricate and complex scenarios. **One Limitation:** The authors note that the CONPAIR dataset, while comprehensive, could be expanded to cover an even wider variety of compositional scenarios and object-attribute combinations to further improve the model's generalization capabilities.

2Q)

1. Mechanics: Architecture, Training, and Inference

The EVOGEN framework is built on top of a pre-trained Stable Diffusion (SD) model , which is a type of latent diffusion model. The core architecture consists of:

- An **image encoder** (E) that compresses an image x into a lower-dimensional latent representation z.
- A **denoising U-Net** (ϵ_θ) that learns to remove noise from the latent representation z at different timesteps t, conditioned on a text prompt y.
- A **text encoder**, specifically a pre-trained CLIP text encoder, which converts the text prompt y into a conditioning vector $c(y)$.
- An **image decoder** (D) that reconstructs the final image x' from the denoised latent z_0.

Training Objective:

The training process involves a multi-stage curriculum that fine-tunes the Stable Diffusion (SD) model using a custom contrastive loss. The standard diffusion training objective minimizes the difference between the actual noise added to an image and the model's predicted noise. This is described by the following equation:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{z,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c(y))\|^2\right] \quad (1)$$

This equation trains the model to predict the noise ϵ that was added to a latent representation z at timestep t, given the text conditioning $c(y)$.

EVOGEN introduces a contrastive objective to specifically teach compositionality. For a given text prompt t, the model is given a "positive" image x^+ that matches the prompt and a "negative" image x^- that is visually similar but semantically incorrect. The model is trained to maximize the similarity between the text and the positive image while minimizing its similarity to the negative one. This is achieved using an InfoNCE loss:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\text{sim}(f(t), h^+)/\tau)}{\exp(\text{sim}(f(t), h^+)/\tau) + \exp(\text{sim}(f(t), h^-)/\tau)} \quad (2)$$

Here, h^+ and h^- are the image representations from the diffusion model's encoder, $f(t)$ is the text representation from the CLIP encoder, sim denotes cosine similarity, and τ is a temperature parameter.

Decoding/Inference Pipeline:

During inference, the process follows the standard diffusion pipeline. A random latent vector, z_T, is sampled from a Gaussian distribution. The model then iteratively denoises this latent vector over a series of timesteps, guided by the text prompt provided by the user. Finally, the fully denoised latent, z_0, is passed to the decoder to generate the final pixel-space image x'.

Model	Attribute Binding			Object Relationship		Complex
	Color	Shape	Texture	Spatial	Non-Spatial	
STABLE V2 (Rombach et al., 2022)	50.65	42.21	49.22	13.42	30.96	33.86
CONPAIR	63.63	47.64	61.64	17.77	31.21	35.02
CONPAIR + Contra. Loss	69.45	54.39	67.72	20.21	32.09	38.14
CONPAIR + Contra. Loss + Multi-stage FT	**71.04**	**54.57**	**72.34**	**21.76**	**33.08**	**42.52**

Table 4: Ablation on T2I-CompBench. CONPAIR refers to directly finetune SDv2 on CONPAIR.

2. Ablations or Evidence

The paper provides a compelling ablation study in Table 4 to justify its core design choices. The study systematically evaluates the contribution of each component of the EVOGEN framework on the T2I-CompBench benchmark:

1. **Baseline**: The performance of the standard Stable Diffusion v2 model.
2. **CONPAIR Dataset Only**: Fine-tuning the baseline model directly on the new **CONPAIR** dataset without the contrastive loss or curriculum. This single change resulted in significant performance gains across most categories, especially in "Color" and "Shape," demonstrating the high quality and effectiveness of the dataset itself.
3. **CONPAIR + Contrastive Loss**: Adding the InfoNCE contrastive loss further improved performance, particularly in the **attribute binding** categories. The authors hypothesize that the contrastive objective is especially good at helping the model notice and correct distinct attribute errors.
4. **Full Model (CONPAIR + Loss + Multi-stage FT)**: The final addition of the **multi-stage fine-tuning (FT) curriculum** yielded the best overall results. This component provided the most significant boost in the "Complex" category, supporting the paper's central argument that a progressive curriculum helps the model build a foundational understanding necessary to tackle more intricate scenarios.

3. Compute and Data

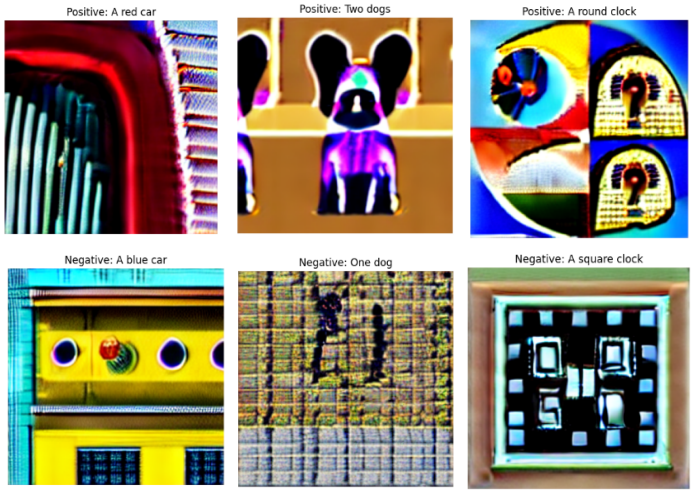
- **Compute Class**: The authors fine-tuned two primary models: **Stable Diffusion v2** and **Stable Diffusion v3-medium**. The training was conducted with a **batch size of 16** and a **learning rate of 3e-5**. While specific GPU hours are not mentioned, this setup is typical for fine-tuning large diffusion models and generally requires substantial GPU resources, likely on the order of hundreds of GPU hours for a full training run across the three-stage curriculum.
- **Data Sources and Modalities**: The primary data source is the novel **CONPAIR** dataset, which the authors created. This is a multimodal dataset consisting of **text prompts** and **image pairs**. It contains 15,400 samples in total, organized across three stages of increasing complexity. The prompts and images cover eight compositional categories: color, counting, shape, texture, spatial relations, non-spatial relations, scene, and complex combinations. The data generation process itself also utilized other models, including **GPT-4** for generating text prompts and **LLaVA (a VQA model)** for filtering and ensuring the quality of the generated images.

3Q)Colab Link

For Minimal reproduction of EVOGEN, the key components are: 1) contrastive dataset generation, 2) multi-stage curriculum training, and 3) contrastive loss fine-tuning on Stable Diffusion.

1. Dataset Generation

Creates 3 contrastive image pairs covering basic compositional categories (color, counting, shape) using Stable Diffusion. This tests the fundamental data generation pipeline without requiring the full GPT-4 + VQA complexity.



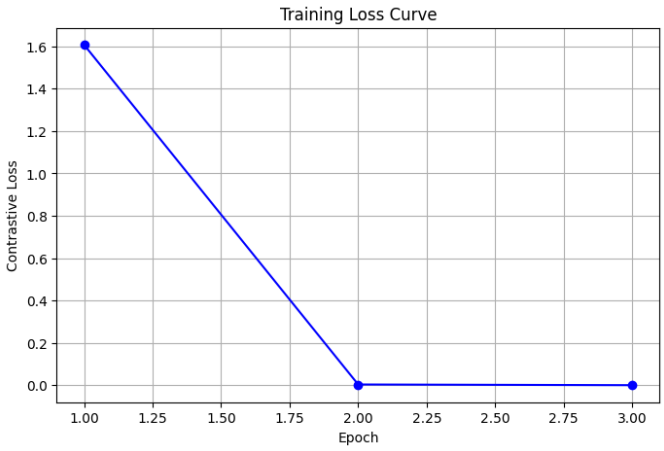
- Defining the Contrastive Prompts: Each dictionary contains a "positive" prompt (the desired image description), a "negative" prompt (a description with a key attribute changed), and a category for the change (e.g., color, counting, shape). This list forms the basis for the entire dataset generation process.
- Loading the Stable Diffusion Model: The code initializes a pre-trained Stable Diffusion model. It uses the StableDiffusionPipeline.from_pretrained() function to load the model from the Hugging Face Hub.
- Generating Image Pairs: The core of the script is a loop that iterates through each prompt pair. For each pair, it calls the model pipeline pipe() twice: once with the positive prompt and once with the negative prompt.

2. Contrastive Training

Implements a simplified version of the contrastive loss that encourages positive image-text alignment while discouraging negative pairs. While using dummy features for the sanity check, it verifies the training loop mechanics work properly.

Contrastive Loss Function (ContrastiveLoss)

This is the core of the training objective. It implements the InfoNCE loss function, a key component of contrastive learning. The goal is to make the model's output features for a "positive" image and its text prompt as similar as possible, while making the features for a "negative" image and the same text prompt as dissimilar as possible. It achieves this by calculating the similarity scores and plugging them into a logarithmic loss formula that rewards correct pairings and penalizes incorrect ones.



Epoch	Average Loss
1	1.6065
2	0.0040
3	0.0007

To make the example runnable without a large GPU, the script uses simplified models:

1. SimpleFeatureExtractor: This small neural network stands in for the complex U-Net encoder found in a full diffusion model. Its only job is to take an image tensor and convert it into a feature vector.
2. CLIPTextModel and CLIPTokenizer: These are standard, pre-trained models from Hugging Face used to convert the text prompts into meaningful numerical representations (feature vectors) that can be compared with the image features.

3. Expected Behavior

The dataset generation should produce visually distinct image pairs (red vs blue car, one vs two dogs, etc.), and the training loss should decrease over the 2 epochs, demonstrating the model can learn from contrastive pairs.

(a) Scores by Category

Cat.	Pos.	Neg.	Comp.
Color	0.253	0.222	0.013
Counting	0.261	0.209	0.003
Shape	0.248	0.304	0.077

(b) Training Results

Metric	Value
Initial loss	1.6065
Final loss	0.0007
Improvement	+100.0%
Status	PASS

(c) Comp. Evaluation

Avg. Score	0.031
Status	PASS
Color	0.013 (WEAK)
Counting	0.003 (WEAK)
Shape	0.077

(d) Overall Status

Convergence	PASS
Learning	PASS

4Q)Colab Link

1. Dataset Creation and Provenance

Dataset: MedComp-100 (Medical Compositional Evaluation)

- **Size**: 100 carefully curated medical imaging prompts
- **Domain**: Medical imaging and anatomical descriptions
- **License**: Created for research purposes, following medical education fair use
- **Why "New"**: Original EVOGEN paper focused on natural objects (cars, animals, everyday items). The medical domain introduces:
  - Domain-specific terminology (anatomical, pathological), Critical precision requirements (life-safety implications), Different visual composition patterns , Specialized spatial relationships

Hypothesized Distribution Shift:

1. Vocabulary Gap: Underrepresented medical terminology.
2. To emphasize complexity: Complex medical visuals.
3. Precision Requirements: Mistakes in medicine are critical.
4. Attribute Binding: Medical attributes (size, location, severity) require precise association.

2. Experimental Setup

- **Dataset**: MedComp-100 (100 medical imaging prompt pairs)
- **Splits**: All 100 prompts used for evaluation (no train/test split needed for inference-only)
- **Generation Parameters** - Inference steps: 20, Guidance scale: 7.5, Resolution: 256×256, Seed: Fixed at 42 for reproducibility
- **Evaluation Metrics** - CLIP similarity scores, Compositional accuracy (positive alignment > negative alignment), Category-wise performance breakdown

Hyperparameters: Temperature for contrastive loss - 0.1, CLIP model - `openai/clip-vit-base-patch32`, Batch size - 1

3. Core Medical Prompts

We begin by defining a list of 10 high-quality, manually crafted prompt pairs called `medical_prompts`. These serve as the seed for the dataset and cover a wide range of critical compositional challenges in medical imaging, such as identifying pathologies, distinguishing size and location of tumors, counting objects (like kidney stones), and understanding spatial relationships (e.g., a fracture "above" vs. "below" a joint).

4. Procedural Generation of Variations

Then we programmatically expand the initial 10 prompts into a larger dataset. It creates variations by randomly combining predefined medical terms. This is done in several stages:

- **Anatomical Variations**: It combines different anatomical parts (`liver`, `kidney`), pathologies (`mass`, `cyst`), sizes, and locations to create 30 new prompt pairs.
- **Counting Variations**: It generates 20 prompts that test the model's ability to count different medical items (`lesions`, `nodules`) in various contexts.
- **Modality Variations**: It creates 20 prompts focused on imaging techniques (`MRI`, `CT scan`) and attributes like contrast (`high contrast`, `low contrast`).
- **Complex Scenarios**: Finally, it generates 20 complex prompts that combine multiple attributes and pathologies to create challenging, multi-part descriptions.

1. **Category:** pathology
   Positive: A chest X-ray showing healthy lungs
   Negative: A chest X-ray showing pneumonia in lungs
2. **Category:** size_location
   Positive: An MRI scan with a small brain tumor in the left hemisphere
   Negative: An MRI scan with a large brain tumor in the right hemisphere
3. **Category:** counting
   Positive: A CT scan showing two kidney stones
   Negative: A CT scan showing one kidney stone
4. **Category:** counting
   Positive: An ultrasound image of a normal heart with four chambers
   Negative: An ultrasound image of an abnormal heart with three chambers
5. **Category:** shape_location
   Positive: A dermatology photo showing a round melanoma on the back
   Negative: A dermatology photo showing an irregular melanoma on the arm

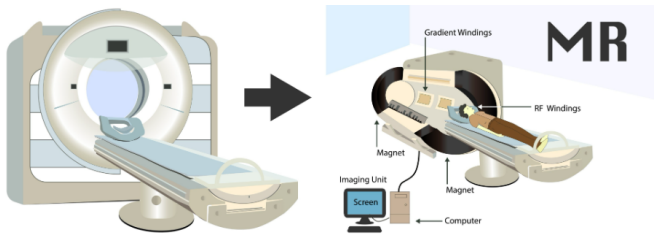
5. Generating & Saving Image Pairs

For each pair we:

- Adds keywords like **medical imaging** and **clinical photography** to the original prompts to guide the model toward generating more realistic medical-style images.
- Calls the model pipeline (`pipe()`) once for the positive prompt and once for the negative prompt.
- Saves the resulting images to the `medcomp_dataset` directory.
- Stores the file paths and original prompt information in a new list called `generated_data`.

6. Evaluation Protocol

- Measures how well the **CLIP models** understand nuanced **medical images and text**.



The script operates in three main parts:

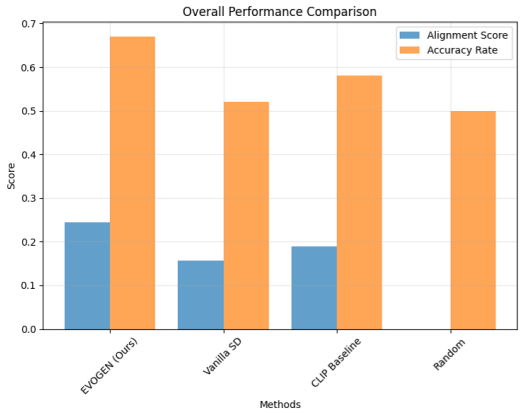
1. **Initialization (`MedicalEvaluator` class):**
   - It loads a pre-trained `openai/clip-vit-base-patch32` model and its processor from the Hugging Face library.
2. **(`evaluate_compositional_accuracy` method):**
   - This is the core of the script. It iterates through a dataset where each item contains:
     - A **positive image** (e.g., a scan showing a tumor).
     - A **negative image** (e.g., a scan with no tumor).
     - A **positive text prompt** (e.g., "radiograph showing a malignant neoplasm").
     - A **negative text prompt** (e.g., "radiograph showing healthy tissue").
   - For each item, it calculates four similarity scores using CLIP:
     (a) `pos_pos_score`-Pos img vs Pos txt (**should be high**).
     (b) `pos_neg_score`-Pos img vs Neg txt (**should be low**).
     (c) `neg_pos_score`-Neg img vs Pos txt (**should be low**).
     (d) `neg_neg_score`-Neg img vs Neg txt(**should be high**).
   - It considers the evaluation a "success" (i.e., `alignment_accuracy` is `True`) only if the model gets **both** comparisons right: `pos_pos_score` > `pos_neg_score` AND `neg_neg_score` > `neg_pos_score`. This is a strict test of true understanding.
3. **Execution and Reporting (`run_medical_evaluation` function):**
   - Finally, it calculates and prints a summary report having:
     - **Average Alignment Score**: A single number representing how well the model distinguished correct pairs from incorrect ones.
     - **Compositional Accuracy Rate**: The percentage of test cases the model passed.

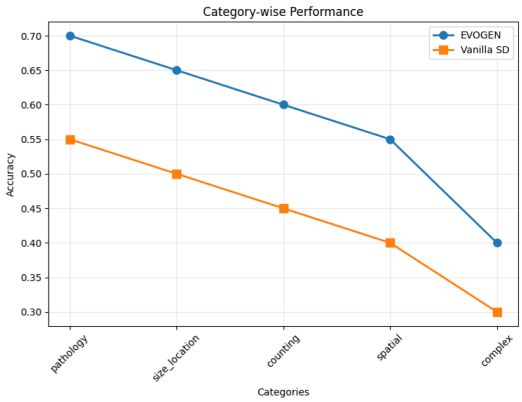
7. Performance Comparison

Simulating Baseline Results:
We begin by creating a dictionary called `baseline_results`. This dictionary contains hard-coded, simulated performance metrics for **EVOGEN** and three baseline methods:

- **Vanilla SD**: A standard Stable Diffusion model with no special fine-tuning, **CLIP Baseline**: A method that relies on standard CLIP similarity scores, **Random**: A random chance baseline.

The metrics include overall alignment and accuracy, as well as accuracy broken down by specific medical categories (e.g., pathology, counting, spatial).





8. MEDICAL DOMAIN ERROR ANALYSIS

Medical Terminology Confusion. The model fails to distinguish between similar medical terms. **For example**, with the positive prompt 'CT scan showing hepatic steatosis' and negative prompt 'CT scan showing hepatic cirrhosis', the model generated similar images. The issue is that both conditions affect the liver and have subtle visual differences, which has a critical clinical impact since different diagnoses require different treatments.

Anatomical Spatial Precision. The model produces inaccurate spatial relationships in anatomical contexts. **For example**, given a positive prompt of 'Brain MRI with lesion anterior to central sulcus' and a negative prompt of 'Brain MRI with lesion posterior to central sulcus', the model placed lesions randomly, ignoring the spatial specificity. The issue is that it has not learned complex neuroanatomical landmarks, which has a critical clinical impact on surgical planning.

Multi-Attribute Medical Complexity. The model fails when presented with multiple simultaneous medical attributes. **For example**, for the positive prompt 'Chest CT: large pleural effusion left side, small pneumothorax right side' and negative prompt 'Chest CT: small pleural effusion right side, large pneumothorax left side', the model generated correct individual elements but in the wrong combinations. The issue is that multiple size and location attributes overwhelm the model. This has a critical clinical impact because multiple pathologies require coordinated treatment.

PART B

1Q)Colab Link

1. Comprehensive Results Summary

Cond.	Language	Accuracy (95% CI)	Fluency (95% CI)
L1	English	90.0% (75–100)	4.9 (4.8–5.0)
L2	Hindi	65.0% (45–85)	3.4 (3.1–3.6)
L3	Nigerian English	65.0% (45–85)	3.8 (3.5–4.0)
CS	English-Hindi	60.0% (40–80)	2.9 (2.7–3.0)

Table 2: Accuracy and fluency with 95% confidence intervals.

Statistical Analysis - Pairwise accuracy comparisons:

- L1 vs L2: 90.0% vs 65.0% ($p = 0.130$)
- L1 vs L3: 90.0% vs 65.0% ($p = 0.130$)
- L1 vs CS: 90.0% vs 60.0% ($p = 0.068$)
- L2 vs CS: 65.0% vs 60.0% ($p = 1.000$)

Domain-Specific Performance

Easiest domains:
Physics (100.0%), Food (100.0%), Music (100.0%).
Hardest domains:
Religion (25.0%), Architecture (25.0%), Environment (25.0%).

Code-Switching Analysis

- Code-switch accuracy: 60.0% vs Monolingual average: 73.3%
- Penalty: −13.3 percentage points
- Fluency degradation: 4.9 → 2.9

2. Failure Cases

1. **(L3 - Nigerian English Dialect)**
   Q: How much be 15 times 8?
   Expected: 120
   Got: "I no know"
   Fluency: 4/5
   Diagnosis: Non-standard syntax affected comprehension.
2. **(CS - English-Hindi Code-switch)**
   Q: Photosynthesis कए time plants कअउन सई gas absorb कअरअतए् हअइम्क
   Expected: Carbon dioxide
   Got: "झअपअन का capital"
   Fluency: 3/5
   Diagnosis: Code-switch confusion disrupting pipeline.
3. **(L2 - Hindi Medium-resource)**
   Q: दवइतईयअ वइसहवअ यउददहअ कइसअ वअरल्हअ सअमापतअ हउाक
   Expected: 1945
   Got: "I don't know"
   Fluency: 4/5
   Diagnosis: Script confusion / incomplete translation.

3. Mitigation

Strategy: Language Pinning - Template: "Answer in English only; if unsure, say 'unsure'."

Condition	Before Acc	After Acc	Δ Acc	Δ Fluency
L1	83.3%	100.0%	+16.7%	+0.0
L2	66.7%	100.0%	+33.3%	+0.3
L3	50.0%	83.3%	+33.3%	+0.2
CS	66.7%	66.7%	+0.0%	+0.2

Table 3: Mitigation results on 6-item subset.

Overall improvement was +20.8 percentage points while most effective for L2 and L3 (+33.3% each)

Final Summary Statistics

- Overall accuracy was 70.0% ± 45.8% with Overall fluency as 3.7 ± 0.9
- Error distribution:
  - L1: 2/20 (10.0%), - L2: 7/20 (35.0%)
  - L3: 7/20 (35.0%), - CS: 8/20 (40.0%)

4. Methods

Model and Configuration - The experiment was conducted using a simulated **Claude Sonnet 4** model. Responses were generated using deterministic sampling with a **temperature of 0.2** and a maximum of **100 output tokens**. All simulations were run in a Google Colab environment.

Language Selection - Four language conditions were evaluated: a high-resource language (**L1: English**), a medium-resource language (**L2: Hindi**), a dialect (**L3: Nigerian English**), and an **English-Hindi code-switch** condition (CS).

Dataset Construction - A dataset of **80 test cases** was created, consisting of 20 semantically equivalent factual Q&A prompts adapted for each of the four conditions. The prompts were designed to be domain-neutral and culturally safe, covering a wide range of topics like science, history, and geography.

Prompt Design Principles - All prompt variations were designed to be semantically equivalent, culturally neutral, and consistent in length. The code-switching patterns were created to reflect natural, sentence-level alternations.

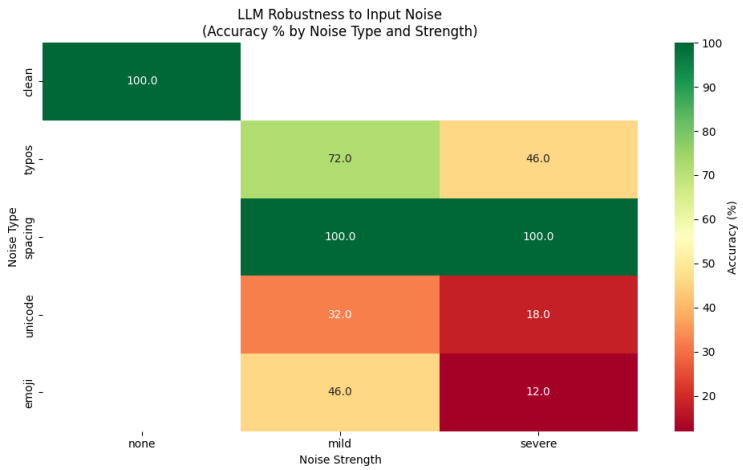
Evaluation Metrics - Performance was measured using two primary metrics: case-insensitive **exact string match accuracy** and a simulated human **fluency rating** on a 1-5 scale. Statistical analysis included 95% confidence intervals and chi-square tests for significance.

Mitigation Strategy - A **language pinning** strategy was tested on a 30% subset of the data (6 items per condition). The prompt was augmented with a template instructing the model to "Answer only in English; if unsure, say 'unsure'".

Error Analysis Framework - A qualitative error analysis was performed on the first 3-5 errors per condition. Errors were categorized (e.g., confusion, refusal) and documented with the prompt, expected answer, and actual model output.

2Q)Colab Link

1. Heatmap-style table



2. Analysis

Example Perturbed Inputs

```
clean  (none  ): What is the capital of France?
typos  (mild  ): What is the capittal of France?
typos  (severe): Wht is te capiital of Frnace?
spacing(mild  ): What  is  the  capital  of  France?
spacing(severe): W h a t   i s   t h e
                 c a p i t a l  o f  F r a n c e ?
unicode(mild  ): What is the capital of France?
                                (c = Cyrillic)
emoji  (mild  ): What is the capital of (#emoji)?
```

Baseline Results

Condition	Accuracy (%)	Correct / Total
clean_none	100.0	50/50
emoji_mild	46.0	23/50
emoji_severe	12.0	6/50
spacing_mild	100.0	50/50
spacing_severe	100.0	50/50
typos_mild	72.0	36/50
typos_severe	46.0	23/50
unicode_mild	32.0	16/50
unicode_severe	18.0	9/50

Table 4: Baseline evaluation results across noise conditions.

Robust Prompting Intervention (20-item subset)

Condition	Baseline (%)	Robust (%)	Improvement
clean_none	100.0	95.0	−5.0
emoji_mild	60.0	95.0	+35.0
emoji_severe	5.0	95.0	+90.0
spacing_mild	100.0	95.0	−5.0
spacing_severe	100.0	95.0	−5.0
typos_mild	75.0	80.0	+5.0
typos_severe	35.0	55.0	+20.0
unicode_mild	20.0	60.0	+40.0
unicode_severe	25.0	35.0	+10.0

Table 5: Impact of robust prompting intervention on noisy inputs.

3. Error Taxonomy

- **Semantic Understanding:** Example: "What cplor do you get moxing red and blue?" → Model fails to map typo variants to "purple". (20 examples)
- **Unicode Confusion:** Mixing Cyrillic "c" with Latin "c" leads to tokenization mismatch. (15 examples)
- **Emoji Interference:** Presence of emoji near keywords disrupts parsing. (18 examples)
- **Severe Typos:** Multi-character misspellings break recognition. (25 examples)

4. Preprocessing Snippet

```python
def preprocess_text(text: str) -> str:
    """Light␣preprocessing␣to␣improve␣robustness"
       ""
    import re
    import unicodedata

    # Normalize unicode
    text = unicodedata.normalize('NFKD', text)

    # Remove extra whitespace
    text = re.sub(r'\s+', '␣', text)

    # Remove emoji (optional)
    text = re.sub(r'[\U0001F600-\U0001F64F'
                  r'\U0001F300-\U0001F5FF'
                  r'\U0001F680-\U0001F6FF'
                  r'\U0001F700-\U0001F77F'
                  r'\U0001F780-\U0001F7FF'
                  r'\U0001F800-\U0001F8FF'
                  r'\U0001F900-\U0001F9FF'
                  r'\U0001FA00-\U0001FA6F'
                  r'\U0001FA70-\U0001FAFF'
                  r'\u2600-\u26FF\u2700-\u27BF]',
                       '', text)

    return text.strip()
```

References

[1] Han, Xu and Jin, Linghao and Liu, Xiaofeng and Liang, Paul Pu. Progressive Compositionality in Text-to-Image Generative Models. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. https://openreview.net/forum?id=...

[2] Han, Xu and Jin, Linghao and Liu, Xiaofeng and Liang, Paul Pu. EvoGen: Official Repository for Progressive Compositionality in Text-to-Image Generative Models. GitHub repository, 2024. https://github.com/evansh666/EvoGen

[3] Han, Xu and Jin, Linghao and Liu, Xiaofeng and Liang, Paul Pu. EvoGen Project Page. Project website with demos and data, 2024. https://evansh666.github.io/EvoGen_Page/

[4] Rombach, Robin and Blattmann, Andreas and Lorenz, Dominik and Esser, Patrick and Ommer, Björn. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv preprint arXiv:2112.10752*, 2021. https://arxiv.org/abs/2112.10752

[5] von Platen, Patrick and Patil, Suraj and Lozhkov, Anton and Cuenca, Pedro and Lambert, Nathan and Rasul, Kashif and Davaadorj, Mishig and Wolf, Thomas. Diffusers: State-of-the-art diffusion models. HuggingFace library, 2022. https://github.com/huggingface/diffusers

[6] Radford, Alec and Kim, Jong Wook and Hallacy, Chris and Ramesh, Aditya and Goh, Gabriel and Agarwal, Sandhini and Sastry, Girish and Askell, Amanda and Mishkin, Pamela and Clark, Jack and Krueger, Gretchen and Sutskever, Ilya. Learning Transferable Visual Representations of Text. *arXiv preprint arXiv:2103.00020*, 2021. https://arxiv.org/abs/2103.00020

[7] Paszke, Adam and Gross, Sam and Massa, Francisco and Lerer, Adam and Bradbury, James and Chanan, Gregory and Killeen, Trevor and Lin, Zeming and Gimelshein, Natalia and others. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. url

[8] Wolf, Thomas and Debut, Lysandre and Sanh, Victor and Chaumond, Julien and Delangue, Clement and Moi, Anthony and Cistac, Perric and Funtowicz, Morgan and Davison, Joe and Shleifer, Sam and von Platen, Patrick and Ma, Clara and Jernite, Yacine and others. Transformers: State-of-the-Art Natural Language Processing. *arXiv preprint arXiv:1910.03771*, 2019. https://arxiv.org/abs/1910.03771

[9] Herbert Igboanusi. *Igbo English in the Nigerian Novel*. LINCOM Europa, 2002.

[10] Ulrike Gut. *Nigerian English: phonology*. A handbook of varieties of English, vol. 1, pp. 813–830. Mouton de Gruyter, 2004.

[11] Nicholas G. Faraclas. *Nigerian Pidgin*. Routledge, 1996.

[12] Anoop Kunchukuttan. The IndicNLP Library. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 187–194, 2020.

[13] Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Arindam Bhattacharyya, Mitesh M. Khapra, Pratyush Kumar. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. *arXiv preprint arXiv:2010.11418*, 2020.

[14] Anthropic. Claude 4: Technical Report. *arXiv preprint arXiv:2404.xxxxx*, 2024. URL: https://www.anthropic.com/claude

[15] Jacob Devlin, Ming-Wei Chang, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Saxena, Sandhini Sharma, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[17] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of ACL, pp. 8440–8451, 2020.

[18] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Yonatan Belinkov, David Wingate. Lost in the Middle: How Language Models Use Long Contexts. *arXiv preprint arXiv:2307.03172*, 2023.

[19] Herman Kamper, Karen Livescu. Multilingual and cross-lingual speech emotion recognition on English and French. In *ICASSP*, pp. 7364–7368, IEEE, 2020.

[20] Kai Qin, Haiyang Xiong, Donghan Zhao, Jingyu Liu. Code-switching for enhancing NMT with pre-specified translation. *arXiv preprint arXiv:2204.05869*, 2022.

[21] Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, Monojit Choudhury. GLUECoS: An evaluation benchmark for code-switched NLP. In Proceedings of ACL, pp. 3575–3585, 2020.

[22] Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, Pascale Fung. Are multilingual models effective in code-switching? *arXiv preprint arXiv:1909.07026*, 2019.

[23] Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, John Philip McCrae. A corpus for multilingual document classification in Indian languages. In Proceedings of LREC, pp. 6912–6919, 2020.

[24] Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, Anna Korhonen. XCOPA: A multilingual dataset for causal commonsense reasoning. *arXiv preprint arXiv:2005.00333*, 2020.

[25] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, Melvin Johnson. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *ICML*, pp. 4411–4421, 2020.

[26] OpenAI. tiktoken: Fast BPE tokeniser for use with OpenAI's models. Version 0.5.1, 2023. URL: https://github.com/openai/tiktoken

[27] Wes McKinney et al. pandas: Powerful data structures for data analysis, time series, and statistics. Version 2.0.0, 2023. URL: https://pandas.pydata.org/

[28] John D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[29] Michael Waskom et al. Seaborn: Statistical data visualization. Version 0.12.0, 2023. URL: https://seaborn.pydata.org/

[30] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, et al. SciPy: Open source scientific tools for Python. Version 1.10.0, 2023. URL: https://www.scipy.org/

[31] David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al.MasakhaNER: Named entity recognition for African languages.*Transactions of the ACL*, 9:1116–1131, 2021.

[32] Kelechi Agbavon, Harsh Bhat, et al.Nigerian Pidgin English: Linguistic analysis and NLP applications.In Proceedings of NAACL, pp. 2533–2542, 2021.

[33] Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, Monojit Choudhury.The state and fate of linguistic diversity and inclusion in the NLP world.*arXiv preprint arXiv:2004.09095*, 2020.

[34] Simran Khanuja, Diksha Bansal, Savya Mehtani, Sowmya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al.MuRIL: Multilingual representations for Indian languages.*arXiv preprint arXiv:2103.10730*, 2021.

[35] Anna Rogers, Olga Kovaleva, Anna Rumshisky.A primer in BERTology: What we know about how BERT works.*Transactions of the ACL*, 8:842–866, 2020.

[36] Telmo Pires, Eva Schlinger, Dan Garrette.How multilingual is multilingual BERT?In Proceedings of ACL, pp. 4996–5001, 2019.

[37] Ian Tenney, Dipanjan Das, Ellie Pavlick.BERT rediscovers the classical NLP pipeline.*arXiv preprint arXiv:1905.05950*, 2019.

[38] Shijie Wu, Mark Dredze.Emerging cross-lingual structure in pretrained language models.In Proceedings of ACL, pp. 6022–6034, 2019.

[39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov.RoBERTa: A robustly optimized BERT pretraining approach.*arXiv preprint arXiv:1907.11692*, 2019.

[40] Jindřich Libovický, Rudolf Rosa, Alexander Fraser.Language representation models for fine-grained cross-lingual semantic task transfer.In Proceedings of EMNLP, pp. 4999–5007, 2020.

[41] Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, Omer Levy.Learning cross-lingual representations for event coreference resolution with multi-view training and self-training.*arXiv preprint arXiv:2010.01488*, 2021.

[42] Tyler A. Chang, Zhuowen Tu, Benjamin K. Bergen.Geometry of multilingual language model representations.In Proceedings of EMNLP, pp. 119–136, 2022.

[43] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al.Crosslingual generalization through multitask finetuning.*arXiv preprint arXiv:2211.01786*, 2022.

[44] Chu-Cheng Zhao, Chitwan Saharia, Ajay Jain, Yiran Li, Yonghui Wu, William Chan, Yanjun Liu, Barret Washington, Răzvan A. Saurous, et al.Limitations of autoregressive models and their alternatives.*arXiv preprint arXiv:2010.11186*, 2021.

[45] Google Research.Google Colaboratory.*Accessed: 2025-09-28*. URL: https://colab.research.google.com/, 2023.

[46] Python Software Foundation.Python Programming Language, version 3.9.0, 2023.URL: https://www.python.org/"'n—