Mauve Scores

1. **Fuel Tank** (Fuel Tank scores are incorrect due to a single sample)
    - Control: 0.0040720962619612555
    - Translator GPT: 0.0040720962619612555
    - JSON GPT: 0.0040720962619612555
2. **Picture Taking System**
    - Control: 1.0
    - Translator: 1.0
    - JSON GPT: 1.0
3. **Camera**
    - Control: 0.9999999999999997
    - Translator: 0.9999999999999997
    - JSON GPT: 0.9999999999999997
4. **Axle System**
    - Control: 1.0
    - Translator: 1.0
    - JSON GPT: 1.0
5. **Mass Requirement**
    - Control: 1.0000000000000004
    - Translator: 1.0000000000000004
    - JSON GPT: 1.0000000000000004
6. **Car with shape using Constructive Solid Geometry (CSG)**
    - Control: 0.9999999999999997
    - Translator: 0.9999999999999997
    - JSON GPT: 0.9999999999999997
7. **Flashlight**
    - Control: 0.999999999999999
    - Translator: 0.99999999999999
    - JSON GPT: 0.9999999999999999
8. **Wheel**
    - Control: 1.0000000000000004
    - Translator: 1.0
    - JSON GPT: 1.0000000000000007
9. **Dynamics Analysis**
    - Control: 0.9999999999999997
    - Translator: 0.9962292026277653
    - JSON GPT: 0.9962292026277653

10. **Analysis Individual**
    - Control: 0.9971431490192941
    - Translator: 1.0000000000000004
    - JSON GPT: 1.0
11. **Cart**
    - Control: 1.0000000000000002
    - Translator: 0.9977609018297389
    - JSON GPT: 0.9988913152976364
12. **Vehicle Hierarchy**
    - Control: 0.9981979186400809
    - Translator: 0.9794668764951738
    - JSON GPT: 0.998850548764123
    -
13. **Surveillance System**
    - Control: 0.9999999999999999
    - Translator: 0.9985184140935454
    - JSON GPT: 0.9990763903254931
14. **Distiller**
    - Control: 0.9999999999999998
    - Translator: 0.9999999999999998
    - JSON GPT: 1.0000000000000004
15. **Room Model**
    - Control: 0.9999999999999998
    - Translator: 0.9999999999999998
    - JSON GPT: 0.9999999999999997
16. **Server Sequence Model**
    - Control: 0.999156290629092
    - Translator: 0.9999999999999999
    - JSON GPT: 0.9999999641028227
17. **Complex Vehicle Variability System**
    - Control: 0.9963554005803621
    - Translator: 0.9999999999999994
    - JSON GPT: 1.0
18. **Vehicle Geometry and Coordinate Frames**
    - Control: 1.0
    - Translator: 0.9999999999999999
    - JSON GPT: 0.9999999999999999

19. **Simple Quadcopter**
    - Control: 1.0
    - Translator: 0.9996486993022586
    - JSON GPT: 1.0
20. **Turbojet Stage Analysis**
    - Control: 1.0000000000000004
    - Translator: 1.0
    - JSON GPT: 0.9999999999999996

➢ From this point on, we will be grading models that exist outside of the knowledge base

21. **HSUV Dynamics**
    - Control: 0.9983176688148692
    - Translator: 0.9998704916641372
    - JSON GPT: 0.9999999999999998

22. **Medical Device Failure**
    - Control: 0.9999999999999998
    - Translator: 0.9999999999999994
    - JSON GPT: 0.9999999999999994
23. **Issue Metadata Example**
    - Control: 0.983417705843693
    - Translator: 1.0
    - JSON GPT: 0.9990660525337609
24. **Vehicle Analysis Demo**
    - Control: 0.9883244986009952
    - Translator: 0.9999999999999997
    - JSON GPT: 0.9876647842297752
25. **EV Sample**
    - Control: 0.9993110441356944
    - Translator: 1.0000000000000004
    - JSON GPT: 0.999311044135694

Proposition: When it comes to simpler models, a fine-tuned GPT performs about the same as an unmodified GPT.

- Simple Cumulative Score (n=8)
  - Control: 1.0
  - SysML Translator: 1.0000000000000007
  - JSON GPT Score: 1.0

➢ From these results, we can ascertain that an unmodified GPT is just as capable of producing an accurate textual output as a fine-tuned GPT, with both models producing results with perfect to near-perfect accuracy.

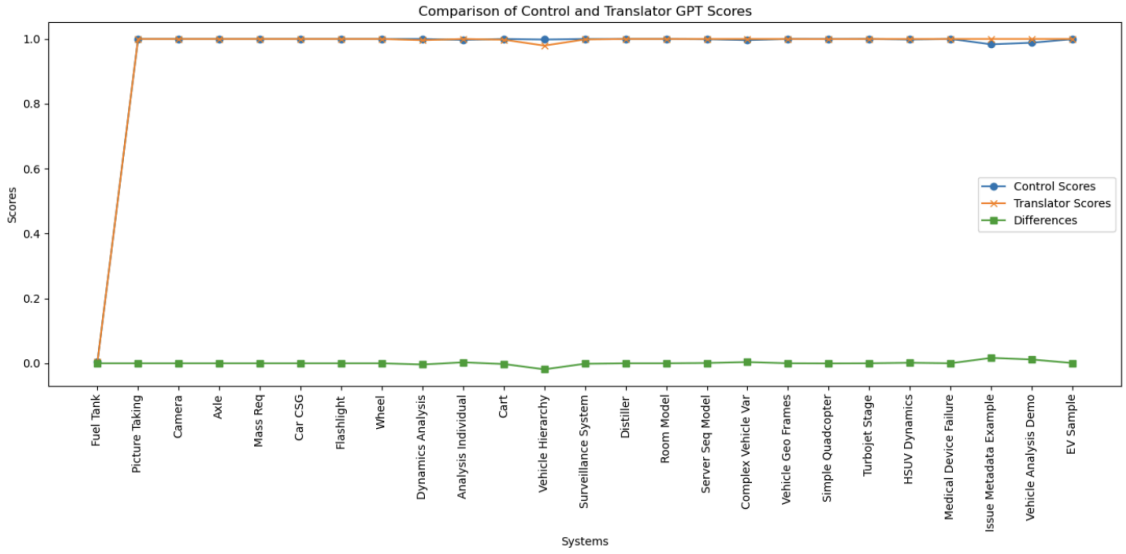
➢ This begs the question, does a fine-tuned GPT perform better when the systems get more complex?

- Complex Cumulative Score (n=17)
  - Control: 0.9851028655309841
  - SysML Translator: 0.9999999999999994
  - JSON GPT Score: 0.9999999999999999

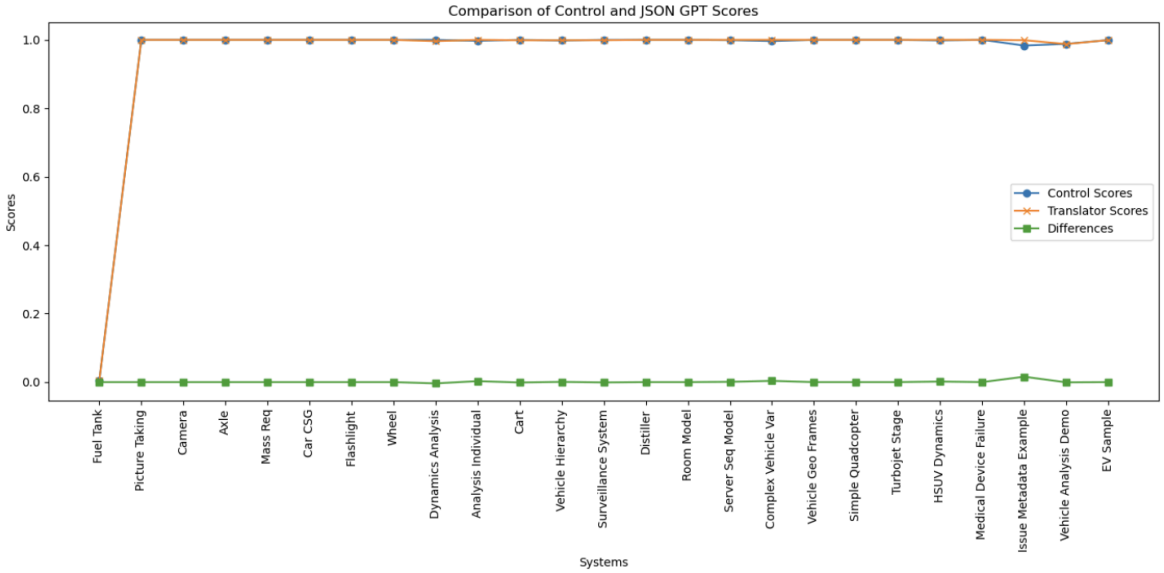
➢ With a fine-tuned GPT outperforming an unmodified GPT by ~1.5%, we note a slight increase in accuracy.

➢ However, since both GPTs posted high accuracy in interpreting SysMLv2 images, we must determine whether this result is statistically significant. To do this, I will perform a paired t-test based on the differences in scores.

```
The p value for the paired t-test is 0.7003 which is far greater than the typical significance level of 0.05.
Therefore we fail to reject our null hypothesis that there is no difference in accuracy between the two GPTs. In conclusion,
there is not enough evidence to suggest that using a fine-tuned GPT will yield better results than an unmodified GPT.

The t statistic was 0.3895 which indicates that the mean difference between the unmodified GPT and fine-tuned GPT scores is
close to zero.
```

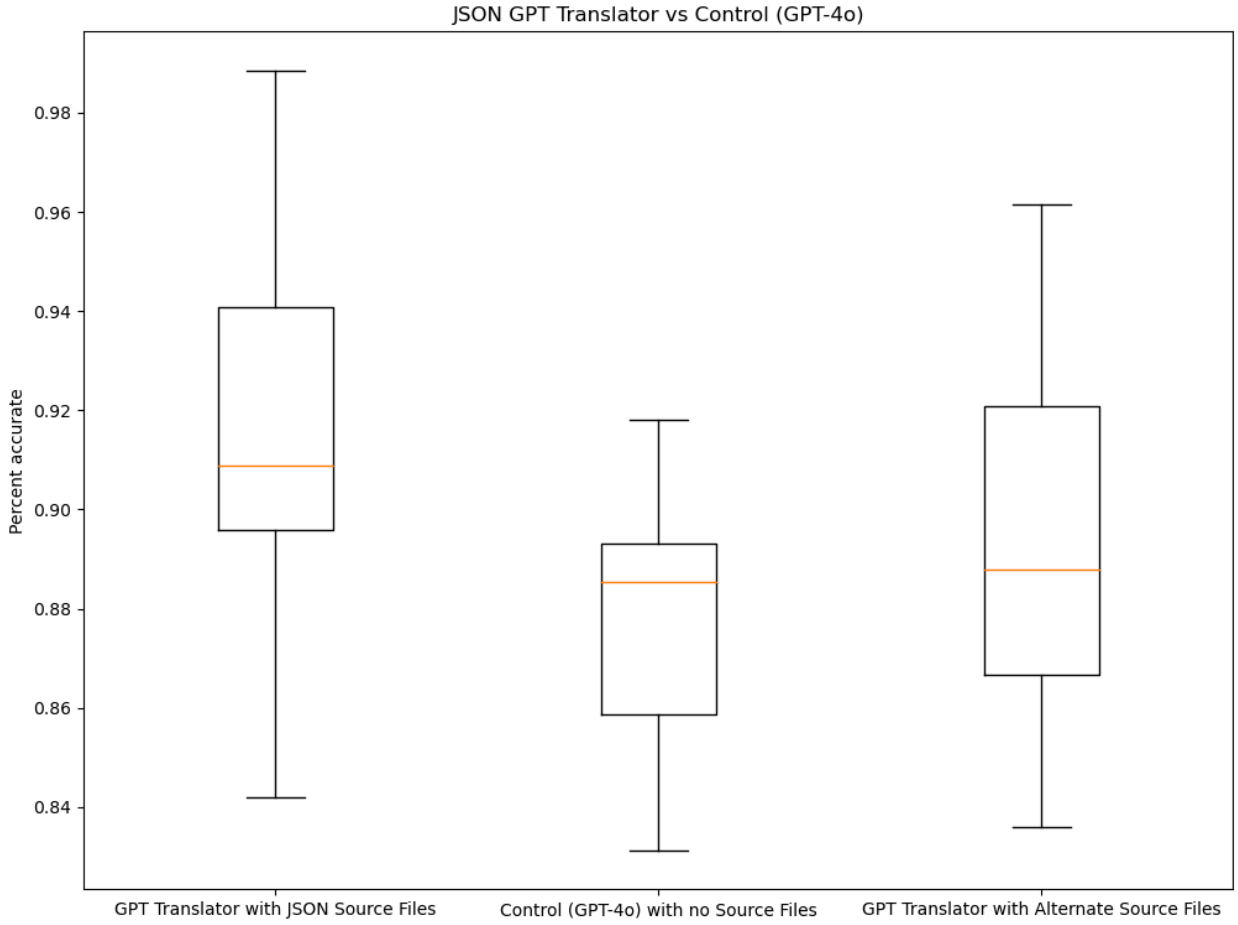


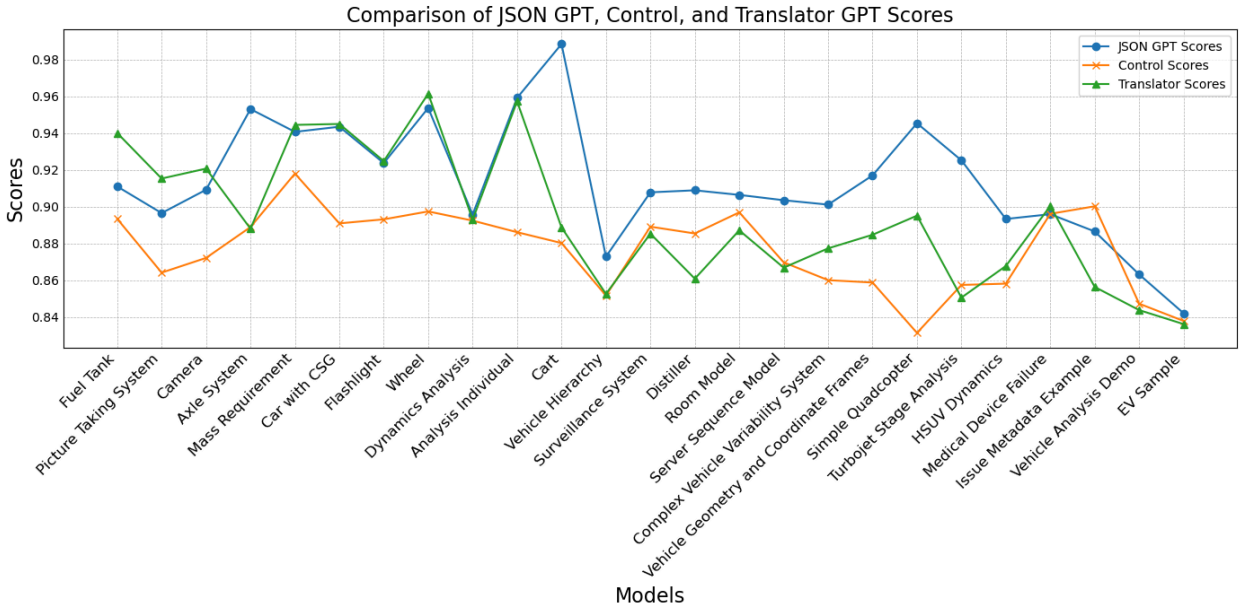
```
The p value for the paired t-test is 0.2741 which is greater than the typical significance level of 0.05.
Therefore we fail to reject our null hypothesis that there is no difference in accuracy between the two GPTs. In conclusion,
there is not enough evidence to suggest that using a fine-tuned GPT will yield better results than an unmodified GPT.

The t statistic was 1.1192 which indicates that the mean difference between the unmodified GPT and fine-tuned GPT scores is
close to one percent.
```

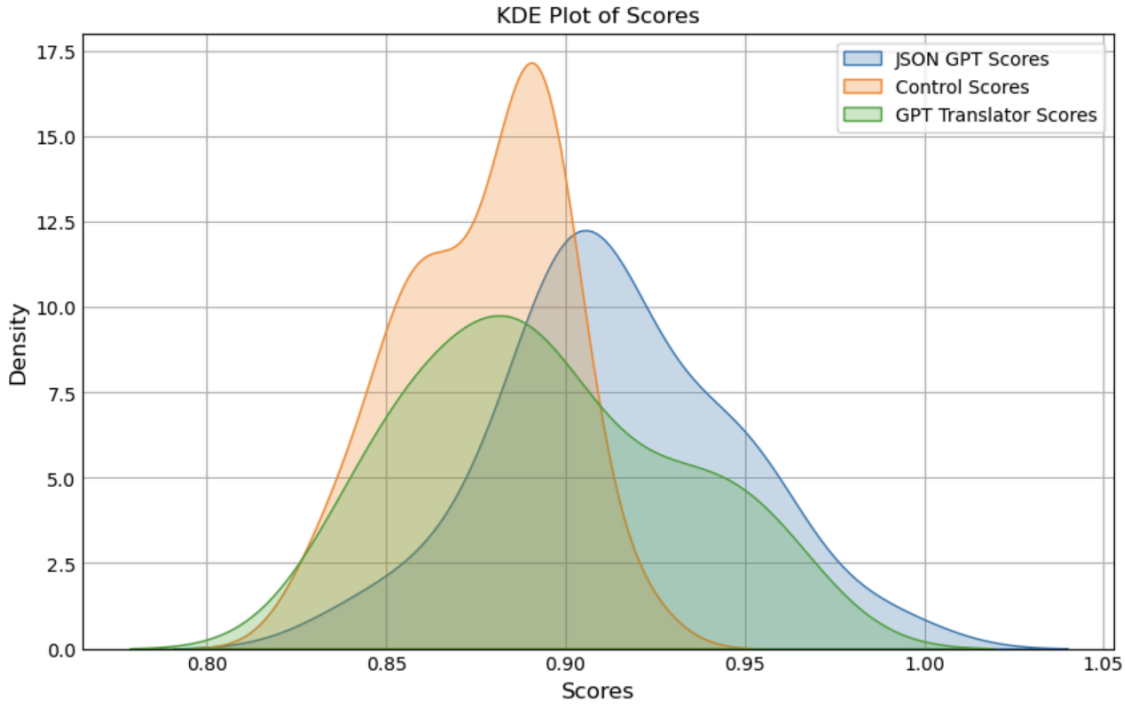


- Although the performance difference between the Control GPT and the fine-tuned GPT was not statistically significant, LLMs still demonstrate a substantial ability to translate SysMLv2 models into textual descriptions. Given their consistent performance, scoring above the 98th percentile in accurately interpreting SysMLv2 models, it is likely that fine-tuning with clear, efficient data further enhances their translation capabilities.

At this point during my investigation on testing the LLMs' ability to convert SysML images to accurate textual descriptions, I felt as if the results I obtained did not pass the eye test. Results from Control GPT did not adequately describe systems in the holistic manner that the actual descriptions did, yet the ratings for the Control GPT were >98% accurate for the most part. I was unable to determine if it was a flaw in the data or the MAUVE scoring system itself since it is robust to so many factors. To address this, I began looking into an alternative grading method, the BERTScore. "BERTScore leverages the pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity. It has been shown to correlate with human judgment on sentence-level and system-level evaluation" (1). While MAUVE uses BERT to convert text into embeddings, I chose to use BERTScore's implementation of the BERT model due to its simplicity and ability to pass the eye test. Using the BERTScore, I found results that moreso aligned with my original hypothesis. I chose to analyze the scores by performing a hypothesis test (two-sample t-test), and the analysis revealed that the JSON GPT model achieved a statistically significantly higher mean score than the other models. This finding supports my initial hypothesis that a fine-tuned GPT would outperform an unmodified GPT in translating SysMLv2 model images to accurate textual descriptions.


JSON GPT Translator vs Control (GPT-4o)
Percent accurate
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
GPT Translator with JSON Source Files
Control (GPT-4o) with no Source Files
GPT Translator with Alternate Source Files


Comparison of JSON GPT, Control, and Translator GPT Scores
JSON GPT Scores
Control Scores
Translator Scores
Scores
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
Fuel Tank
Picture Taking System
Camera
Axle System
Mass Requirement
Car with CSG
Flashlight
Wheel
Dynamics Analysis
Analysis Individual
Cart
Vehicle Hierarchy
Surveillance System
Distiller
Room Model
Server Sequence Model
Complex Vehicle Variability System
Vehicle Geometry and Coordinate Frames
Simple Quadcopter
Turbojet Stage Analysis
HSUV Dynamics
Medical Device Failure
Issue Metadata Example
Vehicle Analysis Demo
EV Sample
Models



The Kernel Density Plot shows that the JSON GPT Scores are concentrated around 0.91.
The JSON GPT scores are concentrated around higher scores than the other GPTs.
While the GPT Translator has a lower density comparatively, it still outperformed the Control GPT.
This is evident looking at the right side of the GPT Translator's distribition.
Instead of falling like a normal distribution, there is a fair amount of data that brings its average score higher than the Control GPT's average score despite having the lowest scores by clustered density.

```python
from scipy import stats
t_stat, p_value = stats.ttest_ind(json_gpt_scores, control_scores)

print(f"JSON GPT vs Control: t-statistic = {t_stat}, p-value = {p_value}\n")


print(f"The p-value of {p_value:.8f} is less than the typical significance level of 0.05 therefore we can reject the null

t_stat, p_value = stats.ttest_ind(gpt_translator_scores, control_scores)

print(f"GPT Translator vs Control: t-statistic = {t_stat}, p-value = {p_value}")

print()

print(f"The p-value of {p_value:.8f} essentially matches than the typical significance level of 0.05 therefore we cannot
```

```
JSON GPT vs Control: t-statistic = 4.717198585581804, p-value = 2.096918071603333e-05

The p-value of 0.00002097 is less than the typical significance level of 0.05 therefore we can reject the null hypothesis
and conclude that there is a statistically significant difference in accurately translating SysMLv2 models to textual
descriptions between a finetuned GPT using JSON source files and an unmodified GPT.

GPT Translator vs Control: t-statistic = 2.0042943503629034, p-value = 0.05069824306663962

The p-value of 0.05069824 essentially matches than the typical significance level of 0.05 therefore we cannot conclude
whether not we can reject the null hypothesis, but we can make the statement that the evidence is trending towards
being statistically significant.
```

References

1. Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, Yoav Artzi. "BERTScore: Evaluating Text Generation with BERT." *International Conference on Learning Representations*, 2020.
2. Sanford Friedenthal, E.S., Roger Burkhart, Eran Gery, Hisashi Miyashita, Hans Peter de Koning, Oystein Haugen, Tomas Juknevicius, Charles Krueger, Ivan Gomes, Miyako Wilson, Santiago Leon, William Piers, Tilo Schreiber, Zoltán Ujhelyi, OMG Systems Modeling Language™. June 2023. p. 638.
3. Sanford Friedenthal, A.M., Rick Steiner, A Practical Guide to SysML : The Systems Modeling Language. 2015, Elsevier/Morgan Kaufmann: Waltham, MA
4. Ed Seidewitz, M.B., SysML-v2-Release. 2024: Github
5. Benjamin Kruse, T.G., Kristina Shea, Martin Eigner, Systematic Comparison of Functional Models in SysML for Design Procedia CIRP, 2014. 21: p. 34-39.