C SATD comments

This appendix presents a examples of selected AD (SATD) comments extracted from Liu et al. [54]'s dataset, supporting the thematic analysis in RQ1 (Definition of AD). The table (11) categorises the comments into primary categories — inefficiency, scalability limitations, model degradation, or unclassified. Each entry includes a rationale to clarify the classification. Readers can refer to this table for a deeper understanding of how the AD categories of inefficiency, scalability limitations, and model degradation were derived, complementing the findings in Section 4.1.

Table 11. List of Selected Studies

S/N	Comment	Primary Category	Rationale
1	TODO: Add direct conversion, since creating an intermediate array might be very slow	Inefficiency	Relates to performance optimisation in code efficiency.
2	TODO: batchSize is fixed to one. Needs to find out how to handle batch axis as a free dimension	Scalability Limitations	Fixed batch size limits large-scale processing.
3	TODO: Set NUM lattice to null to save memory.	Inefficiency	Memory optimization suggestion.
4	TODO: Add an epsilon [CR comment by Nikos]	Model Degradation	Addresses numerical stability, risking accuracy loss.
5	BUGBUG: Can one pass a NumPy array as initial values? TODO: Add a test case.	Unclassified	Bug and test focus, unclear AD impact.
6	Convolution – create a convolution layer with optional non-linearity	Unclassified	A design feature request, not directly related to TD.
7	TODO: Can we specify atrous (dilated) convolution? How?	Unclassified	A request for a feature, not related to TD.
8	TODO: sharing = false?	Unclassified	A configuration bug, not affecting TD.
9	TODO: Conflict of parameter order: filter_shape or num_filters first?	Unclassified	Bug in configuration order, not TD.
10	TODO: Stride not supported for sequential.	Unclassified	A feature request, not related to TD.
11	TODO: Should we cater to the special case of 1D convolution for text? I.e. sequential only (filter_shape=()).In that case, the convolution is the embedding, and we should use a matrix product to support sparse inputs.Or add sparse support to splice().	Scalability Limitations	The comment suggests adapting the convolution process for text processing efficiency and optimising sparse input handling, which can impact scalability in large-scale NLP models.

(Continued from previous page)

S/N	Comment	Primary Category	Rationale
12	TODO: Dynamic axis for NumPy arrays.	Inefficiency	A code improvement suggestion to increase flexibility.
13	TODO: Sparse for NumPy arrays.	Inefficiency	A suggestion for enhancing efficiency in handling sparse data.
14	The version info for the project you're documenting acts as a replacement for version and...	Unclassified	Pertains to project documentation, not a TD issue.
15	TODO: The current implementation is O(N^2), it should be possible to do this in O(N logN).	Scalability Limitations	Performance scaling issue related to algorithm complexity.
16	scaling up minibatch_size increases sample throughput. In 8-GPU machine,ResNet110 samples-per-second is $\tilde{7}$x of single GPU, comparing to $\tilde{3}$x without scalingup. However, bigger minibatch size on the same number of samples means less updates,thus leads to higher training error. This is a trade-off of speed and accuracy	Model degradatgion	Increasing the mini-batch size improves throughput but reduces update frequency, leading to higher training error, which impacts model accuracy.
17	TODO: Express using Delay() layer.	Unclassified	A request for an implementation feature, not TD.
18	TODO: Is there a better way to discriminate?	Model Degradation	Improving the model's discriminative power.
19	TODO: Test this—useful for implementing recursions over arrays without having to pass the number around.	Unclassified	Code improvement related to recursion handling.
20	Use two spaces for better visual separability.	Unclassified	Related to formatting and readability, not TD.
21	Default quantiser is short, symmetric.	Unclassified	Performance issue regarding quantiser implementation.
22	Model of late 2015, which had a bug in setting InputValue's tensor dimensions.	Unclassified	Historical bug, unclear current AD impact.
23	TODO: Should not wait, simply publishing an event on the compute stream should be sufficient.	Inefficiency	Optimization suggestion to improve event handling.

(Continued from previous page)

S/N	Comment	Primary Category	Rationale
24	TODO: Should slice off the supplied Value object instead of reallocating.	Inefficiency	Suggesting a more efficient approach for object handling.
25	TODO: Except X, all other inputs to LSTM are treated as constant.	Model Degradation	Suggestion for model architecture improvement.
26	TODO: In principle, all these variables shall be treated as either constant or op output.	Model Degradation	Improving model variable handling.
27	TODO: Make bidirectional LSTM work by figuring out output data layout transpose.	Model degradation	This could relate to AD if it impacts the correctness of LSTM output.
28	TODO: Or should we just blast m_distanceToStart to GPU, and mask based on that?	Inefficiency	Suggestion to improve GPU handling for performance.
29	TODO: Should both activations be replaced? bit = it * activation(bit_proj) applied to tanh of input network	Model Degradation	Affects model architecture and performance.