Response to Reviewers

Manuscript: Graph Enhanced Transformer for Semi-Supervised Duplicate Bug Report Detection
Authors: Kerem Bayramoğlu, Hüseyin Karaca, Emirhan Koç, Hasan Erkin Ünlü, Rabia Ela Ünlü

We sincerely thank all reviewers for their thorough and constructive feedback. We have carefully addressed each comment and revised the manuscript accordingly. Below, we provide detailed responses to each reviewer's comments and indicate the corresponding changes in the revised manuscript.

Summary: We have carefully addressed all reviewer comments through manuscript revisions, expanded discussions, and acknowledgments of limitations. Key changes include: (1) substantial revision of Section 2 for clearer problem definition and motivation, (2) comprehensive grammatical and formatting corrections, (3) expanded Future Work with detailed discussions of experimental directions, (4) explicit acknowledgment of statistical validation limitations, (5) clarification of competitive performance value, and (6) improved reproducibility package. We believe these revisions have significantly strengthened the manuscript.

Reviewer 1

Overall Recommendation: Weak Accept

We thank the reviewer for the careful review and positive comments, particularly regarding the strong baseline comparisons and effective use of unlabeled data.

Comment 1.1

In Section 5.4 (lines 955–970), the paper discusses the training process by only mentioning starting and ending loss values. It would be more helpful to reference the figures before starting the discussion or include loss curves at the same page to improve reading flow.

Response: We thank the reviewer for this valuable suggestion. We agree that explicitly referencing the convergence figures improves both clarity and flow. We have revised the Training Dynamics Analysis section to include explicit references to Figures 3 and 4 at the beginning of the discussion.

Revisions in Manuscript:

↪ **Section 5.4, Page 10:**
"The detailed convergence plots are shown in Figures 3 and 4 for Eclipse and Thunderbird datasets, respectively."

Comment 1.2

Potential improvements include experimenting with pretrained embeddings or learned node features instead of one-hot encodings, testing different PCA dimensionalities, and conducting ablations for hyperparameters such as projection dimension D, fusion weight λ, margin m, and training epochs.

Response: We thank the reviewer for these thoughtful suggestions. We fully acknowledge the value of these ablation studies and sensitivity analyses. Due to computational resource and time constraints, we were unable to conduct comprehensive hyperparameter exploration in this revision. However, we have now explicitly discussed these directions in the expanded Future Work section.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"Several hyperparameters in our framework were fixed based on preliminary experiments without exhaustive ablation studies. The PCA dimensionality (d=10), projection dimension (D=128), fusion weight (λ), margin (m) in the loss function, and number of training epochs all warrant systematic sensitivity analysis. In particular, the aggressive dimensionality reduction from 768 to 10 dimensions via PCA was chosen empirically to improve discrimination but may introduce information loss. Future work should explore different values of d (e.g., 5, 20, 50, 100) and characterize the trade-off between noise reduction and information preservation across different datasets. Such ablation studies would provide data-driven justification for hyperparameter choices and reveal the robustness of the proposed framework."

Comment 1.3

Grammatical issues: GCN used before definition, keyword capitalization, "ie." format, duplicated "for example for example", missing commas, and "token id's" formatting.

Response: We thank the reviewer for bringing these issues to our attention. We have systematically corrected all mentioned errors throughout the manuscript.

Revisions in Manuscript:

↪ **Section , Page 2:**
"Graph Convolutional Network (GCN)"

↪ ***Section , Page 1:***
"Changed "Issue" to "Bug Report" and properly capitalized "Semi-Supervised Learning" for consistency with standard terminology."

↪ ***Section 1, Page 3:***
"for example, 20% of reports in Eclipse and 30% in Firefox were marked as duplicate"

↪ ***Section 4, Page 5:***
"i.e., the negative pair sampling"

↪ ***Section 3.4, Page 5:***
"In Figure 1, corrected the formatting from "token id's" to "token IDs" following standard conventions."

Comment 1.4

The replication package is missing key files and the training notebook shows interrupted execution.

Response: We thank the reviewer for this observation. We have updated our GitHub repository to include all necessary files for complete reproducibility and clarified the repository pointer in the manuscript.

Revisions in Manuscript:

↪ **Section 5.2.2, Page 8:**
"All source code and the full reproducibility package, including all necessary data files and scripts, can be found in the huseyin-karaca/graph-enhanced-dbd GitHub repository 🔗."

Reviewer 2

Overall Recommendation: Weak Reject

We thank the reviewer for the thorough evaluation and constructive criticism.

Comment 2.1

The paper claims to address "label-scarce" environments but uses full datasets. Experiments with reduced training set sizes (5% or 10% of labeled data) are needed.

Response: We thank the reviewer for this important observation. We acknowledge that our current experiments do not directly validate performance under extreme label scarcity. While the graph-based framework is designed to leverage unlabeled data, we agree that experiments with systematically reduced label budgets would provide stronger evidence. Due to time and computational constraints, we could not complete these experiments for this revision. We have added this explicitly to the Future Work section.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"While the current experiments use full training datasets, a key motivation for graph-based semi-supervised learning is its potential effectiveness under label-scarce conditions. Future work should include experiments with reduced training set sizes (e.g., 5% or 10% of labeled data) to empirically validate the claim that graph propagation provides concrete benefits when annotations are limited. Additionally, while we have characterized training-time overhead, future work should include detailed inference latency measurements (e.g., milliseconds per query) comparing the proposed method against baselines, providing quantitative evidence for the inference scalability claims made in RQ2."

Comment 2.2

Table 4 reports only training time overhead. Inference latency metrics must be provided to support scalability claims.

Response: We thank the reviewer for this valuable point. We agree that inference latency measurements would strengthen RQ2. We have clarified that the GNN is discarded at inference time and added quantitative inference benchmarking as future work.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"While the current experiments use full training datasets, a key motivation for graph-based semi-supervised learning is its potential effectiveness under label-scarce conditions. Future work should include experiments with reduced training set sizes (e.g., 5% or 10% of labeled data) to empirically validate the claim that graph propagation provides concrete benefits when annotations are limited. Additionally, while we have characterized training-time overhead, future work should include detailed inference latency measurements (e.g., milliseconds per query) comparing the proposed method against baselines, providing quantitative evidence for the inference scalability claims made in RQ2."

Comment 2.3

The PCA dimensionality reduction from 768 to d=10 appears arbitrary. Sensitivity analysis should be conducted.

Response: We thank the reviewer for raising this concern. We acknowledge that d=10 was empirically motivated but not systematically validated. Due to computational constraints, we have expanded the Future Work section to explicitly highlight the need for PCA dimensionality sensitivity analysis.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"Several hyperparameters in our framework were fixed based on preliminary experiments without exhaustive ablation studies. The PCA dimensionality (d=10), projection dimension (D=128), fusion weight (λ), margin (m) in the loss function, and number of training epochs all warrant systematic sensitivity analysis. In particular, the aggressive dimensionality reduction from 768 to 10 dimensions via PCA was chosen empirically to improve discrimination but may introduce information loss. Future work should explore different values of d (e.g., 5, 20, 50, 100) and characterize the trade-off between noise reduction and information preservation across different datasets. Such ablation studies would provide data-driven justification for hyperparameter choices and reveal the robustness of the proposed framework."

Comment 2.4

Table 5 shows the method provides no concrete benefit over baselines; results are identical or slightly worse.

Response: We thank the reviewer for this observation. We respectfully emphasize that our goal is to demonstrate that a novel graph-enhanced semi-supervised framework can achieve *competitive* performance while explicitly leveraging unlabeled data. Achieving F1 scores that match or closely approximate state-of-the-art models validates the architectural concept. We have added a paragraph emphasizing this perspective.

Revisions in Manuscript:

↪ **Section 5.3.3, Page 10:**
"It is important to emphasize that the goal of this work is not merely to achieve marginal numerical improvements over existing baselines, but rather to demonstrate that a graph-enhanced semi-supervised framework can reach competitive performance while explicitly leveraging unlabeled data during training. The fact that our approach matches or closely approximates the performance of heavily-optimized, fully-supervised transformer models while incorporating structural information from unlabeled reports represents a meaningful contribution. This validates the architectural concept and establishes that graph-based semi-supervised learning is a viable path forward for duplicate bug report detection in label-scarce settings, even if the current instantiation does not universally outperform all baselines."

Comment 2.5

While the GNN application shows originality, similar hybrid architectures exist in other NLP domains, limiting foundational novelty.

Response: We thank the reviewer for this fair assessment. We acknowledge that hybrid architectures have been explored elsewhere. However, to the best of our knowledge, this work represents an early exploration of graph-based semi-supervised learning specifically for duplicate bug report detection, with distinct domain characteristics and design choices.

Revisions in Manuscript: No change was required in the manuscript.

Comment 2.6

The paper is well-written. However, "for example" is duplicated in Line 120.

Response: We thank the reviewer for the positive feedback. We have corrected the duplicated phrase.

Revisions in Manuscript:

↪ **Section 1, Page 3:**
"for example, 20% of reports in Eclipse and 30% in Firefox were marked as duplicate"

Reviewer 3

Overall Recommendation: Weak Reject

We thank the reviewer for the detailed evaluation and constructive criticism.

Comment 3.1

Graph edges use only titles. Ignoring descriptions may discard important relationships.

Response: We thank the reviewer for this insightful observation. We fully agree that descriptions could yield richer structures. Our choice was motivated by computational efficiency. We have expanded Future Work to discuss incorporating descriptions.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"In this work, graph edges are constructed based solely on semantic similarity of bug report titles. While titles provide concise summaries, they are often brief, vague, and incomplete compared to full descriptions. Incorporating description text when computing semantic similarity for edge formation could yield substantially richer relational structures. However, this introduces computational challenges (longer sequences, higher memory requirements) and potential noise (descriptions may contain less relevant information). Future work should explore hybrid approaches that weight title and description similarity, or use multi-view graph construction that creates separate edge types based on different text fields."

Comment 3.2

PCA reduction from 768 to 10 dimensions is very aggressive and may not generalize.

Response: We thank the reviewer for this valid concern. We acknowledge the aggressive reduction and its generalization risks. We have expanded Future Work to highlight the need for PCA dimensionality ablation studies.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"Several hyperparameters in our framework were fixed based on preliminary experiments without exhaustive ablation studies. The PCA dimensionality (d=10), projection dimension (D=128), fusion weight (λ), margin (m) in the loss function, and number of training epochs all warrant systematic sensitivity analysis. In particular, the aggressive dimensionality reduction from 768 to 10 dimensions via PCA was chosen empirically to improve discrimination but may introduce information loss. Future work should explore different values of d (e.g., 5, 20, 50, 100) and characterize the trade-off between noise reduction and information preservation across different datasets. Such ablation studies would provide data-driven justification for hyperparameter choices and reveal the robustness of the proposed framework."

Comment 3.3

Negative sampling may create mostly easy negatives. Hard negative mining could improve decision boundaries.

Response: We thank the reviewer for this point. While our approach combines two strategies to ensure balance, we acknowledge that explicit hard negative mining could further sharpen boundaries. We have added this to Future Work.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"The current negative sampling strategy combines anchor-based pairing with random sampling across duplicate groups. While this ensures balanced representation of positive and negative examples, it may over-represent easy negatives—report pairs that are clearly dissimilar. Hard negative mining, which focuses on difficult non-duplicates (reports that appear similar but describe different issues), could improve the model's ability to learn fine-grained decision boundaries. Future work should explore curriculum-based training strategies that progressively introduce harder negatives, or employ similarity-based negative sampling to deliberately select challenging negative pairs."

Comment 3.4

Maintaining the full graph in memory may become difficult as repositories grow.

Response: We thank the reviewer for this practical concern. We clarify that the GNN is used only during training, not inference. For even larger repositories, we discuss strategies in the expanded Future Work section.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"For extremely large bug repositories (e.g., hundreds of thousands or millions of reports), maintaining the full graph in memory on a single GPU becomes impractical. Future work should investigate distributed computing strategies for both graph construction and GNN training. Techniques such as graph partitioning across multiple GPUs or machines, distributed message passing frameworks (e.g., DistDGL, PyTorch Geometric distributed), and out-of-core graph storage could enable scaling to industrial-scale repositories. Additionally, approximate methods such as graph sampling or mini-batch GNN training on subgraphs could reduce memory footprint while maintaining representational quality."

Comment 3.5

Evaluation limited to Eclipse and Thunderbird constrains generalization claims.

Response: We thank the reviewer for this valid observation. We acknowledge this limitation and have listed evaluation on diverse datasets as future work.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"Our evaluation is limited to two benchmark datasets from large, open-source projects (Eclipse and Thunderbird). These repositories may exhibit similar reporting cultures and technical domains. To assess the generalizability of the proposed framework, future work should evaluate performance on a more diverse set of repositories, including smaller projects, proprietary software systems, and domains with different bug reporting guidelines (e.g., mobile applications, embedded systems, web services). Cross-domain evaluation would reveal whether the graph-enhanced approach is robust to variations in vocabulary, reporting style, and duplicate patterns."

Comment 3.6

Overlap between Introduction, Problem Description, and Related Work sections.

Response: We thank the reviewer for this structural feedback. We have substantially revised Section 2 to focus on problem definition and motivation, reducing redundancy.

Revisions in Manuscript:

↪ Section 2, Page 4:

"The core challenge in Duplicate Bug Report Detection (DBRD) is to automatically identify whether two bug reports describe the same underlying issue. In formal terms, given a pair of reports (r_i, r_j), the task is to predict whether they belong to the same duplicate group. This problem is inherently difficult due to lexical variations, incomplete descriptions, and domain-specific terminology that may differ across reporters.

Transformer-based sentence encoders, such as BERT and its variants, have demonstrated strong semantic representations for text matching tasks. However, these models are inherently limited in their ability to exploit unlabeled data beyond implicit pretraining on general-domain corpora. In practical duplicate bug report detection settings, this limitation becomes critical: labeled duplicate pairs are scarce and expensive to obtain, while the majority of available bug reports remain unlabeled. Consider a repository with tens of thousands of bug reports where only a few hundred duplicate relationships have been manually confirmed. Traditional supervised learning approaches would discard the vast majority of this data, utilizing only the small labeled subset.

Directly incorporating all unlabeled bug reports into transformer-based pairwise training is computationally infeasible and methodologically ill-defined. The number of potential report pairs grows quadratically with repository size, making exhaustive pairwise training intractable. Moreover, supervision is only available for a small subset of report pairs, leaving the vast majority of possible comparisons without explicit labels. Existing transformer-based approaches attempt to address this through negative sampling strategies, but these methods do not provide a principled mechanism to leverage the full unlabeled corpus during training.

Motivation for Graph-Based Semi-Supervised Learning. *Graph neural networks offer a natural solution to this challenge. Unlike pairwise supervised learning, GNNs operate on relational neighborhoods and can propagate information across connected nodes without requiring explicit labels for every connection. This property enables us to construct a unified graph representation in which all available bug reports—both labeled and unlabeled—participate in the learning process. Edges can be formed based on label-independent semantic relationships (e.g., title similarity), enabling efficient graph construction without additional annotations.*

In our framework, the transformer component operates on a restricted but feasible subset of bug report pairs during training. Positive pairs are formed from known duplicate bug reports, while negative pairs are generated by sampling from different duplicate groups. The representations learned from these labeled pairs are injected into corresponding graph nodes, and the GNN propagates this information across the entire graph through message passing. Although direct supervision is applied only to nodes participating in labeled pairs, the graph structure allows information to flow to unpaired and unlabeled nodes, enabling them to indirectly contribute to representation learning.

This design decouples pairwise supervision from global data utilization: while the transformer is trained on a manageable subset of labeled and pseudo-labeled pairs, the graph component enables the model to benefit from the full unlabeled corpus. The proposed framework thus bridges the gap between transformer-based semantic modeling and large-scale semi-supervised learning, making it more suitable for realistic, label-scarce duplicate bug report detection settings."

Comment 3.7

Typos and awkward phrases, such as "for example for example".

Response: We thank the reviewer. We have corrected the duplicated phrase and reviewed the manuscript for similar issues.

Revisions in Manuscript:

↪ **Section 1, Page 3:**
"for example, 20% of reports in Eclipse and 30% in Firefox were marked as duplicate"

Comment 3.8

Results don't beat baselines; lack of cross-validation or statistical tests makes small differences unreliable.

Response: We thank the reviewer. We acknowledge both concerns: (1) our goal is competitive performance with unlabeled data leverage, and (2) lack of statistical rigor is a limitation. We have acknowledged this in Threats to Validity.

Revisions in Manuscript:

↪ **Section 5.3.3, Page 10:**
"It is important to emphasize that the goal of this work is not merely to achieve marginal numerical improvements over existing baselines, but rather to demonstrate that a graph-enhanced semi-supervised framework can reach competitive performance while explicitly leveraging unlabeled data during training. The fact that our approach matches or closely approximates the performance of heavily-optimized, fully-supervised transformer models while incorporating structural information from unlabeled reports represents a meaningful contribution. This validates the architectural concept and establishes that graph-based semi-supervised learning is a viable path forward for duplicate bug report detection in label-scarce settings, even if the current instantiation does not universally outperform all baselines."

↪ ***Section 6.1, Page 13:***
"Future work should include rigorous statistical validation through multiple random seeds, cross-validation, or paired significance tests to establish confidence in the observed performance differences."

Reviewer 4

Overall Recommendation: Weak Accept

We thank the reviewer for appreciating the technical rigor and practical applicability of our work.

Comment 4.1

Section 2 reads like methodology rather than problem definition and motivation.

Response: We thank the reviewer for this structural critique. We have substantially revised Section 2 to focus on problem definition and technical motivation.

Revisions in Manuscript:

↪ Section 2, Page 4:

"The core challenge in Duplicate Bug Report Detection (DBRD) is to automatically identify whether two bug reports describe the same underlying issue. In formal terms, given a pair of reports (r_i, r_j), the task is to predict whether they belong to the same duplicate group. This problem is inherently difficult due to lexical variations, incomplete descriptions, and domain-specific terminology that may differ across reporters.

Transformer-based sentence encoders, such as BERT and its variants, have demonstrated strong semantic representations for text matching tasks. However, these models are inherently limited in their ability to exploit unlabeled data beyond implicit pretraining on general-domain corpora. In practical duplicate bug report detection settings, this limitation becomes critical: labeled duplicate pairs are scarce and expensive to obtain, while the majority of available bug reports remain unlabeled. Consider a repository with tens of thousands of bug reports where only a few hundred duplicate relationships have been manually confirmed. Traditional supervised learning approaches would discard the vast majority of this data, utilizing only the small labeled subset.

Directly incorporating all unlabeled bug reports into transformer-based pairwise training is computationally infeasible and methodologically ill-defined. The number of potential report pairs grows quadratically with repository size, making exhaustive pairwise training intractable. Moreover, supervision is only available for a small subset of report pairs, leaving the vast majority of possible comparisons without explicit labels. Existing transformer-based approaches attempt to address this through negative sampling strategies, but these methods do not provide a principled mechanism to leverage the full unlabeled corpus during training.

Motivation for Graph-Based Semi-Supervised Learning. *Graph neural networks offer a natural solution to this challenge. Unlike pairwise supervised learning, GNNs operate on relational neighborhoods and can propagate information across connected nodes without requiring explicit labels for every connection. This property enables us to construct a unified graph representation in which all available bug reports—both labeled and unlabeled—participate in the learning process. Edges can be formed based on label-independent semantic relationships (e.g., title similarity), enabling efficient graph construction without additional annotations.*

In our framework, the transformer component operates on a restricted but feasible subset of bug report pairs during training. Positive pairs are formed from known duplicate bug reports, while negative pairs are generated by sampling from different duplicate groups. The representations learned from these labeled pairs are injected into corresponding graph nodes, and the GNN propagates this information across the entire graph through message passing. Although direct supervision is applied only to nodes participating in labeled pairs, the graph structure allows information to flow to unpaired and unlabeled nodes, enabling them to indirectly contribute to representation learning.

This design decouples pairwise supervision from global data utilization: while the transformer is trained on a manageable subset of labeled and pseudo-labeled pairs, the graph component enables the model to benefit from the full unlabeled corpus. The proposed framework thus bridges the gap between transformer-based semantic modeling and large-scale semi-supervised learning, making it more suitable for realistic, label-scarce duplicate bug report detection settings."

Comment 4.2

Graph construction ignores descriptions, which may forgo significant data.

Response: We thank the reviewer. We acknowledge this limitation and have listed incorporating descriptions as future work.

Revisions in Manuscript:

↪ **Section 6, Page 12:**
"In this work, graph edges are constructed based solely on semantic similarity of bug report titles. While titles provide concise summaries, they are often brief, vague, and incomplete compared to full descriptions. Incorporating description text when computing semantic similarity for edge formation could yield substantially richer relational structures. However, this introduces computational challenges (longer sequences, higher memory requirements) and potential noise (descriptions may contain less relevant information). Future work should explore hybrid approaches that weight title and description similarity, or use multi-view graph construction that creates separate edge types based on different text fields."

Comments 4.3–4.10

Various formatting and terminological issues throughout the manuscript.

Response: We thank the reviewer for meticulous attention to detail. We have systematically corrected all mentioned issues.

Revisions in Manuscript:

↪ **Section , Page 2:**
"Graph Convolutional Network (GCN)"

↪ ***Section , Page 1:***
"Changed "Issue" to "Bug Report" and properly capitalized "Semi-Supervised Learning" for consistency with standard terminology."

↪ ***Section 1, Page 3:***
"for example, 20% of reports in Eclipse and 30% in Firefox were marked as duplicate"

Comment 4.11

Section 2 should include concrete examples of LLM failures and properly describe the DBRD problem.

Response: We thank the reviewer. We have substantially revised Section 2 to include formal problem definition and concrete motivation with realistic scenarios.

Revisions in Manuscript:

↪ Section 2, Page 4:

"The core challenge in Duplicate Bug Report Detection (DBRD) is to automatically identify whether two bug reports describe the same underlying issue. In formal terms, given a pair of reports (r_i, r_j), the task is to predict whether they belong to the same duplicate group. This problem is inherently difficult due to lexical variations, incomplete descriptions, and domain-specific terminology that may differ across reporters.

Transformer-based sentence encoders, such as BERT and its variants, have demonstrated strong semantic representations for text matching tasks. However, these models are inherently limited in their ability to exploit unlabeled data beyond implicit pretraining on general-domain corpora. In practical duplicate bug report detection settings, this limitation becomes critical: labeled duplicate pairs are scarce and expensive to obtain, while the majority of available bug reports remain unlabeled. Consider a repository with tens of thousands of bug reports where only a few hundred duplicate relationships have been manually confirmed. Traditional supervised learning approaches would discard the vast majority of this data, utilizing only the small labeled subset.

Directly incorporating all unlabeled bug reports into transformer-based pairwise training is computationally infeasible and methodologically ill-defined. The number of potential report pairs grows quadratically with repository size, making exhaustive pairwise training intractable. Moreover, supervision is only available for a small subset of report pairs, leaving the vast majority of possible comparisons without explicit labels. Existing transformer-based approaches attempt to address this through negative sampling strategies, but these methods do not provide a principled mechanism to leverage the full unlabeled corpus during training.

Motivation for Graph-Based Semi-Supervised Learning. *Graph neural networks offer a natural solution to this challenge. Unlike pairwise supervised learning, GNNs operate on relational neighborhoods and can propagate information across connected nodes without requiring explicit labels for every connection. This property enables us to construct a unified graph representation in which all available bug reports—both labeled and unlabeled—participate in the learning process. Edges can be formed based on label-independent semantic relationships (e.g., title similarity), enabling efficient graph construction without additional annotations.*

In our framework, the transformer component operates on a restricted but feasible subset of bug report pairs during training. Positive pairs are formed from known duplicate bug reports, while negative pairs are generated by sampling from different duplicate groups. The representations learned from these labeled pairs are injected into corresponding graph nodes, and the GNN propagates this information across the entire graph through message passing. Although direct supervision is applied only to nodes participating in labeled pairs, the graph structure allows information to flow to unpaired and unlabeled nodes, enabling them to indirectly contribute to representation learning.

This design decouples pairwise supervision from global data utilization: while the transformer is trained on a manageable subset of labeled and pseudo-labeled pairs, the graph component enables the model to benefit from the full unlabeled corpus. The proposed framework thus bridges the gap between transformer-based semantic modeling and large-scale semi-supervised learning, making it more suitable for realistic, label-scarce duplicate bug report detection settings."

↪ Section 2, Page 4:

"The core challenge in Duplicate Bug Report Detection (DBRD) is to automatically identify whether two bug reports describe the same underlying issue. In formal terms, given a pair of reports (r_i, r_j), the task is to predict whether they belong to the same duplicate group. This problem is inherently difficult due to lexical variations, incomplete descriptions, and domain-specific terminology that may differ across reporters.

Transformer-based sentence encoders, such as BERT and its variants, have demonstrated strong semantic representations for text matching tasks. However, these models are inherently limited in their ability to exploit unlabeled data beyond implicit pretraining on general-domain corpora. In practical duplicate bug report detection settings, this limitation becomes critical: labeled duplicate pairs are scarce and expensive to obtain, while the majority of available bug reports remain unlabeled. Consider a repository with tens of thousands of bug reports where only a few hundred duplicate relationships have been manually confirmed. Traditional supervised learning approaches would discard the vast majority of this data, utilizing only the small labeled subset.

Directly incorporating all unlabeled bug reports into transformer-based pairwise training is computationally infeasible and methodologically ill-defined. The number of potential report pairs grows

Comments 4.12–4.28

Various additional issues: abbreviations, phrasing, figure references, early stopping details, and scalability discussions.

Response: We thank the reviewer for the comprehensive list. We have systematically addressed all issues including clarifying early stopping criterion, adding figure references, and discussing distributed strategies.

Revisions in Manuscript:

↪ **Section 4, Page 5:**
"i.e., the negative pair sampling"

↪ ***Section 3.4, Page 5:***
"In Figure 1, corrected the formatting from "token id's" to "token IDs" following standard conventions."

↪ ***Section 5.4, Page 10:***
"The detailed convergence plots are shown in Figures 3 and 4 for Eclipse and Thunderbird datasets, respectively."

↪ ***Section 5.2.2, Page 8:***
"early stopping based on validation performance; specifically, training is stopped if validation loss does not improve for 3 consecutive epochs (patience=3)"

↪ ***Section 5.2.2, Page 8:***
"All source code and the full reproducibility package, including all necessary data files and scripts, can be found in the huseyin-karaca/graph-enhanced-dbd GitHub repository 🔗."

↪ ***Section 6, Page 12:***
"For extremely large bug repositories (e.g., hundreds of thousands or millions of reports), maintaining the full graph in memory on a single GPU becomes impractical. Future work should investigate distributed computing strategies for both graph construction and GNN training. Techniques such as graph partitioning across multiple GPUs or machines, distributed message passing frameworks (e.g., DistDGL, PyTorch Geometric distributed), and out-of-core graph storage could enable scaling to industrial-scale repositories. Additionally, approximate methods such as graph sampling or mini-batch GNN training on subgraphs could reduce memory footprint while maintaining representational quality."