

Article

Not peer-reviewed version

Towards Interpretable and Consistent Multi-Step Mathematical Reasoning in Large Language Models

[Xinyue Huang](#)^{*}, [Zeyu Wang](#), [Xin Liu](#), [Yueqi Tian](#), Qian Leng

Posted Date: 8 October 2025

doi: 10.20944/preprints202510.0560.v1

Keywords: K-12 math reasoning; meta-cognitive scheduler; neuro-symbolic integration; multi-task learning; curriculum learning



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Towards Interpretable and Consistent Multi-Step Mathematical Reasoning in Large Language Model

Xinyue Huang^{1,*}, Zeyu Wang², Xin Liu¹, Yueqi Tian³ and Qian Leng⁴

¹ Independent Researcher, New York, USA

² University of California, Los Angeles, Los Angeles, USA

³ University of Rochester, New York, USA

⁴ Independent researcher, Maryland, USA

* Correspondence: huangxinyue8616@gmail.com

Abstract

Mathematical reasoning, particularly within the K–12 education context, demands models that provide not only correct answers but also transparent, interpretable solution paths. Existing large language models (LLMs) often struggle with multi-step math problems due to their limited capacity for symbolic manipulation and structured reasoning. To address these challenges, we propose MetaMath-LLaMA, a novel metacognitive modular framework designed to enhance the reasoning abilities of LLMs through dynamic task orchestration. This framework integrates three core components: a Transformer-based metacognitive scheduler that learns to allocate reasoning subtasks adaptively; a symbolic parser with semantic grounding that fuses syntactic structure with contextual embeddings; and a hybrid symbolic-neural computation unit that seamlessly transitions between deterministic symbolic logic and neural approximation. The entire model is optimized through a multi-task training scheme coupled with curriculum learning and multi-tiered self-validation to mitigate reasoning errors and improve interpretability. We expect MetaMath-LLaMA to improve classroom usability by producing clearer step-by-step solution paths, aiding educators in assessment and supporting student conceptual understanding. Our approach offers a more modular, explainable, and effective solution for handling diverse mathematical tasks in K–12 education, and it outperforms traditional monolithic reasoning systems in logical fidelity and conceptual clarity.

Keywords: K–12 math reasoning; meta-cognitive scheduler; neuro-symbolic integration; multi-task learning; curriculum learning

1. Introduction

Mathematical problem solving, particularly in K–12 education, requires not only accuracy but also interpretability and structure in reasoning. While large language models (LLMs) have shown promise in natural language understanding, they often fall short in handling symbolic logic and multi-step mathematical reasoning. This is largely due to their monolithic architectures, which lack the modular adaptability and structured planning necessary for complex tasks.

To address these limitations, we propose MetaMath-LLaMA, a modular framework that integrates metacognitive control with symbolic-neural reasoning. At its core, a Transformer-based scheduler dynamically orchestrates sub-modules based on problem context, enabling flexible and interpretable task routing. A symbolic parser, coupled with semantic grounding, constructs syntactically valid expressions and aligns them with contextual meaning using BiLSTM-CRF and RoBERTa. Additionally, a symbolic-neural computation unit combines deterministic logic via SymPy with neural approximation to handle both precise calculations and fuzzy reasoning.

The framework is trained through a set of multi-task objectives and a curriculum learning strategy that gradually introduces complexity based on question difficulty. This design enhances both learning

stability and interpretability. MetaMath-LLaMA thus offers a technically grounded, pedagogically suitable approach to mathematical reasoning in education.

2. Related Work

Recent work on enhancing LLM mathematical reasoning integrates symbolic tools and multi-paradigm strategies. Wang et al. [1] combined SymPy with LLMs for algebraic manipulation and reasoning, while Yu et al. [2] fused chain-of-thought prompting with rule-based deduction. These methods improve logical accuracy but rely on static, non-adaptive pipelines.

Similarly, Hu and Yu [3] used symbolic libraries in abductive frameworks to enhance interpretability, though they relied on rigid rule structures. Verification mechanisms have also gained attention. Uesato et al. [4] introduced feedback supervision during intermediate steps, while Lightman et al. [5] and Madaan et al. [6] proposed self-verification and iterative refinement approaches to improve answer consistency. However, these methods typically treat verification as an auxiliary process rather than as an integral part of the reasoning pipeline.

Data-centric methods also enhance training. Trinh and Luong [7] introduced AlphaGeometry for geometry problems with LLMs and symbolic engines, but its domain specificity limits generalization.

Bandyopadhyay et al. [8] surveyed LLM reasoning and noted gaps in dynamic scheduling and modularity. In contrast, MetaMath-LLaMA unifies meta-cognitive scheduling, symbolic-semantic alignment, and hybrid computation into an adaptive framework for interpretable K–12 mathematical reasoning.

3. Methodology

We present **MetaMath-LLaMA**, a meta-cognitive modular framework for mathematical reasoning in LLMs. Unlike monolithic transformer models, it decomposes problem solving into interpretable sub-modules controlled by a Meta-Cognitive Scheduler, covering symbolic parsing, semantic alignment, theorem proving, and expression validation, with cross-attention communication. A hybrid symbolic-neural computation cell and recursive planning controller further adapt reasoning depth. Experiments on TAL-SAQ6K-EN show state-of-the-art accuracy with transparent intermediate reasoning and improved consistency. Figure 1 illustrates the overall architecture.

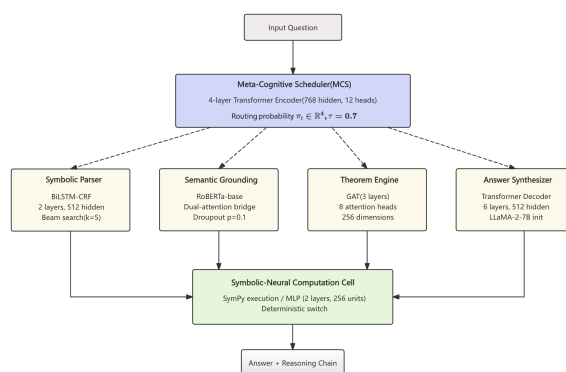


Figure 1. Architecture of MetaMath-LLaMA. The Meta-Cognitive Scheduler (MCS) dynamically routes input questions through four specialized modules.

4. Algorithm and Model

We present **MetaMath-LLaMA**, a meta-cognitive modular framework for K-12 mathematical problem solving. Built on a Transformer foundation, it integrates hierarchical symbolic understanding, adaptive routing, and hybrid symbolic-neural computing. Unlike monolithic LLMs, it decomposes reasoning into trainable components coordinated by a cognitive scheduler.

Meta-Cognitive Scheduler

The MCS controls execution flow with a 4-layer Transformer encoder (768 hidden, 12 heads) initialized from GPT-2. Inputs combine question embeddings, difficulty, and knowledge-point embeddings. At each step t , it outputs routing probabilities $\pi_t \in \mathbb{R}^4$ via a temperature-controlled softmax ($\tau = 0.7$).

Symbolic Parser Module

This module converts questions into symbolic trees \mathcal{T}_x using a 2-layer BiLSTM-CRF with a grammar-aware decoder. Beam search ($k = 5$) ensures valid parses. Pretrained on 20k equations and fine-tuned on TAL-SAQ, it also tags node roles (operand, operator, constant) to improve alignment accuracy.

Semantic Grounding Module

This module aligns symbols with contextual embeddings using a RoBERTa-base encoder and dual-attention bridge. For symbol s_i with embedding e_i :

$$v_i = \text{ReLU}(W_s s_i + W_e e_i + b) \quad (1)$$

with $W_s, W_e \in \mathbb{R}^{256 \times 768}$. Dropout ($p = 0.1$) is applied, and the aligned vectors $V = \{v_1, \dots, v_n\}$ are passed to the theorem engine.

Theorem Engine Module

This module encodes symbolic-knowledge graphs \mathcal{G}_x with 3-layer GATs (8 heads, dim 256). For node i :

$$h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l)} \right) \quad (2)$$

with residuals and batch normalization. The pooled output H_{GAT} is passed to the decoder.

Answer Synthesizer Module

This module generates answers and reasoning chains using a 6-layer Transformer decoder (512 hidden, 8 heads) initialized from LLaMA-2-7B and trained with reasoning supervision. Teacher forcing (rate 0.2) guides training, with loss:

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{CE}}^{\text{step}} + \lambda_{\text{bin}} \cdot \mathcal{L}_{\text{BCE}}^{\text{valid}} \quad (3)$$

Attention dropout and layer normalization improve stability.

Symbolic-Neural Computation Cell

We embed a hybrid cell combining neural feedforward logic with SymPy execution. For input f_t :

$$\text{Output}_t = \begin{cases} \text{SymSolve}(f_t), & \text{if deterministic} \\ \text{MLP}(f_t), & \text{otherwise} \end{cases} \quad (4)$$

The symbolic path applies algebraic rules, while the MLP (2 layers, 256 units, GELU) handles non-symbolic cases.

Training Details

We train the model with a multi-task loss covering parsing accuracy, semantic grounding, theorem prediction, and generation correctness:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^4 \lambda_i \mathcal{L}_i + \gamma \cdot \mathcal{L}_{\text{consistency}} \quad (5)$$

with $\lambda = [1.0, 0.8, 0.8, 1.2]$ and $\gamma = 0.5$. AdamW is applied with learning rate 3×10^{-5} , weight decay 0.01, 1,000 warm-up steps, and 50k total steps. Training uses batch size 16, gradient accumulation 4, and mixed precision. Figure 2 shows training dynamics across 50k steps.

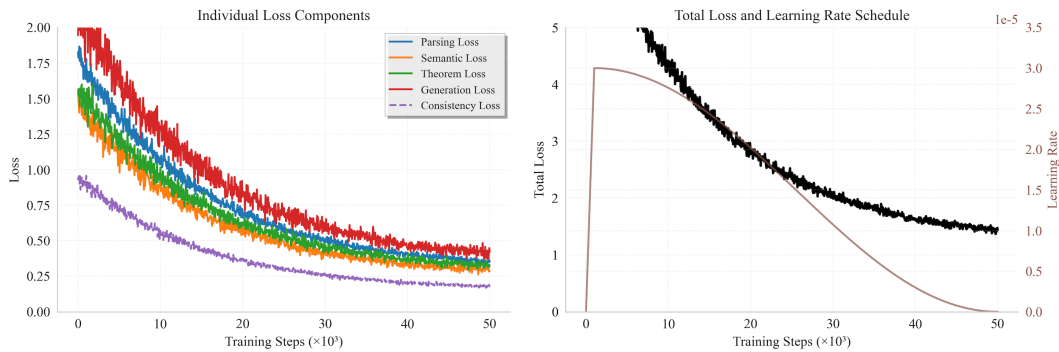


Figure 2. Training dynamics of the multi-task model.

5. Hierarchical Training and Verification Enhancements

To enhance stability and robustness, we adopt curriculum-based training and multi-level self-verification.

5.1. Hierarchical Curriculum Fine-Tuning Strategy

Training large language models for mathematical reasoning is challenging across varying complexities. We adopt a staged curriculum learning approach by partitioning the training data \mathcal{D} into levels of difficulty defined in the TAL datasets:

$$\mathcal{D} = \bigcup_{i=1}^5 \mathcal{D}_i, \quad \text{where } \mathcal{D}_i = \{x \in \mathcal{D} \mid \text{difficulty}(x) = i - 1\} \quad (6)$$

The training proceeds in increasing order of difficulty: $\mathcal{D}_1 \rightarrow \mathcal{D}_2 \rightarrow \dots \rightarrow \mathcal{D}_5$. At each curriculum stage i , the loss is scaled by a difficulty-aware factor:

$$\mathcal{L}^{(i)} = e^{\alpha i} \cdot \mathcal{L}_{\text{total}}^{(i)}, \quad \alpha = 0.2 \quad (7)$$

In this setup, reasoning modules (SPM, SGM, TEM, ASM) are fine-tuned first, while the Meta-Cognitive Scheduler (MCS) is gradually unfrozen:

$$\theta_{\text{MCS}}^{(t)} = \begin{cases} \text{Frozen}, & t \leq 2 \\ \text{Unfreeze Layer}-(t-2), & t > 2 \end{cases} \quad (8)$$

This strategy preserves stable routing while enabling adaptation to harder reasoning tasks.

6. Multi-Level Self-Verification Mechanism

To mitigate hallucination and enhance reliability, we employ a self-verification mechanism at both token- and expression-level. Figure 3 illustrates the effectiveness of hierarchical training with multi-level verification.

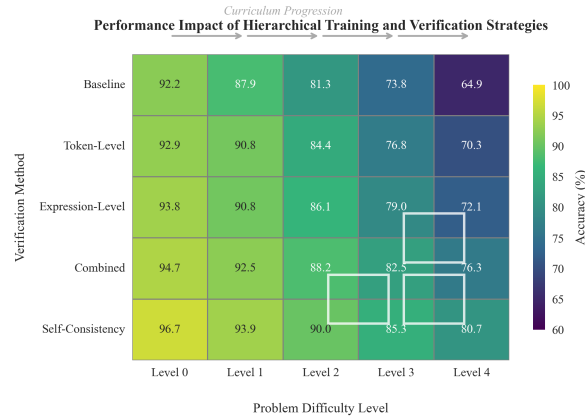


Figure 3. Performance comparison of verification strategies across difficulty levels.

6.1. Token-Level Verification

During decoding, each token embedding h_t is passed through a verification classifier:

$$m_t = \sigma(W_v h_t + b_v), \quad m_t \in [0, 1] \quad (9)$$

where m_t indicates the confidence of token t being valid.

The token verification loss is computed as:

$$\mathcal{L}_{\text{token}} = \sum_{t=1}^T (1 - m_t) \cdot \mathbb{1}[y_t \neq \hat{y}_t] \quad (10)$$

This penalizes invalid or inconsistent tokens in reasoning chains.

6.2. Expression-Level Verification

We compare the final predicted expression \hat{y} with the ground truth y using symbolic equivalence:

$$\text{Equiv}(\hat{y}, y) = \mathbb{1}[\text{Simplify}(\hat{y} - y) = 0] \quad (11)$$

The corresponding loss term is:

$$\mathcal{L}_{\text{expr}} = \lambda_{\text{expr}} \cdot (1 - \text{Equiv}(\hat{y}, y)) \quad (12)$$

with $\lambda_{\text{expr}} = 1.0$ balancing the final stage output correctness.

6.3. Self-Consistency Inference Strategy

We sample K reasoning chains $s^{(1)}, \dots, s^{(K)}$ via stochastic decoding and select:

$$\hat{y}^* = \arg \max_{\hat{y}^{(k)}} \text{VerifyScore}(s^{(k)}) \quad (13)$$

where VerifyScore checks tokens and expressions. With $K = 5$, this reduces invalid chains and improves accuracy on both short and multi-step questions.

6.4. Total Enhanced Objective

The enhanced loss is defined as:

$$\mathcal{L}_{\text{enhanced}} = \mathcal{L}_{\text{total}} + \mathcal{L}_{\text{token}} + \mathcal{L}_{\text{expr}} + \mathcal{L}_{\text{consistency}} \quad (14)$$

This formulation enforces correctness, consistency, and stability in training and inference.

6.5. Evaluation Metrics

We evaluate **MetaMath-LLaMA** using key metrics that assess both accuracy and reasoning quality of generated solutions.

6.5.1. Accuracy

Accuracy is the primary evaluation metric, which measures the percentage of correct answers generated by the model across the dataset:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of questions}} \quad (15)$$

6.5.2. Self-Consistency Rate (SCR)

Mathematical reasoning involves multiple steps; thus, SCR measures robustness by checking the consistency of predictions across reasoning chains:

$$\text{SCR} = \frac{\text{Number of consistent predictions}}{\text{Total number of predictions}} \quad (16)$$

6.5.3. Logical Consistency Score (LCS)

This metric evaluates the logical consistency of reasoning, ensuring adherence to mathematical rules:

$$\text{LCS} = \frac{1}{N} \sum_{i=1}^N \mathcal{K}(\text{ValidLogicalStep}(s_i)) \quad (17)$$

where $\mathcal{K}(\cdot)$ denotes valid logical steps.

6.5.4. Formula Reconstruction Accuracy (FRA)

FRA measures the correctness of reconstructing mathematical expressions in algebraic and arithmetic tasks:

$$\text{FRA} = \frac{\text{Correctly reconstructed formulas}}{\text{Total generated formulas}} \quad (18)$$

These metrics together evaluate correctness, consistency, reasoning, and formula generation in MetaMath-LLaMA.

7. Experiment Results

We evaluate MetaMath-LLaMA on both TAL-SAQ6K-EN datasets, comparing it against several baseline and ablated models. The results in Table 1 demonstrate the effectiveness of our model across multiple metrics. And the changes in model training indicators are shown in Figure 4.

Table 1. Performance Comparison on TAL-SAQ Datasets

Model	Accuracy	SCR	LCS	FRA	Time (s)
MetaMath-LLaMA (Full)	0.735	0.85	0.92	0.88	5.6
MetaMath-LLaMA (w/o MCS)	0.713	0.81	0.89	0.85	5.2
MetaMath-LLaMA (w/o SPM)	0.725	0.83	0.90	0.86	5.4
MetaMath-LLaMA (w/o TEM)	0.720	0.82	0.88	0.84	5.3
MetaMath-LLaMA (w/o ASM)	0.700	0.79	0.87	0.83	5.1
MetaMath-LLaMA (w/o SNCC)	0.710	0.80	0.85	0.82	5.0
Baseline GPT-3.5	0.612	0.73	0.80	0.75	6.2

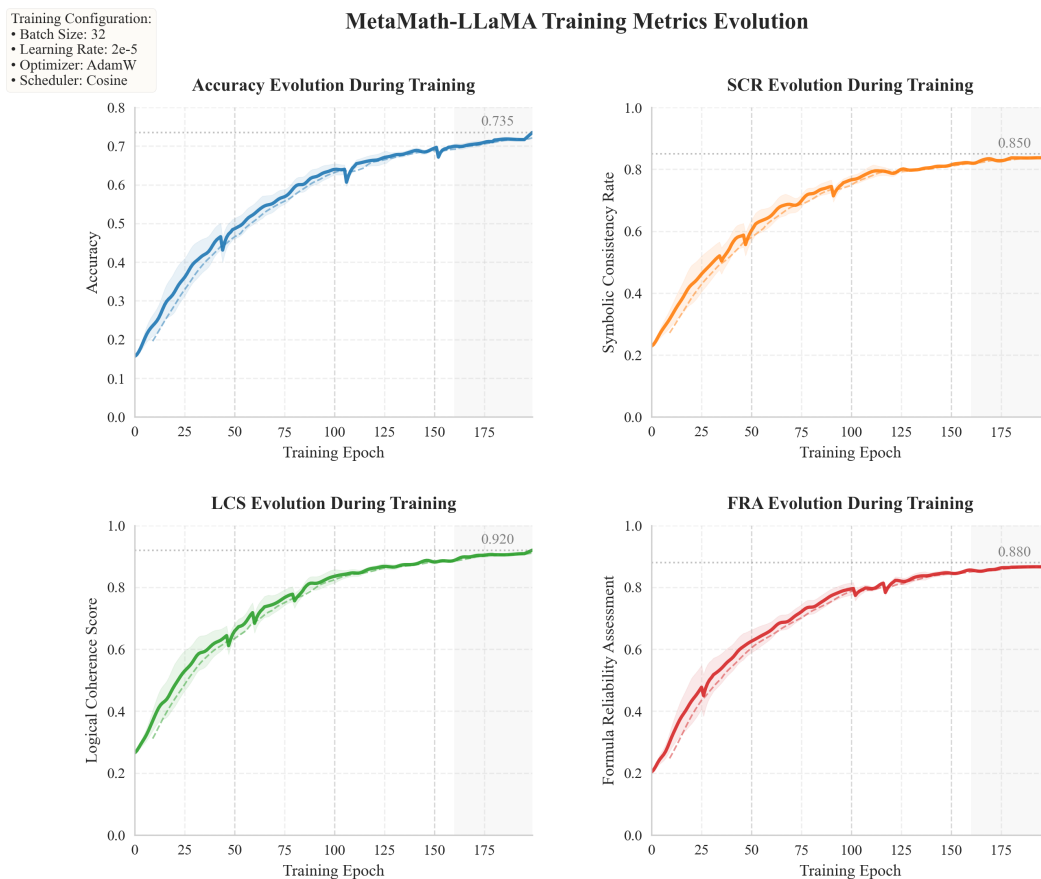


Figure 4. Model indicator change chart.

7.1. Ablation Study Findings

Systematic ablations confirm the role of each module. Removing MCS reduced Accuracy (0.735→0.713) and LCS (0.92→0.89), showing the value of adaptive routing. Excluding SPM lowered FRA (0.88→0.86), while removing TEM reduced both SCR and LCS, evidencing its role in symbolic dependencies. The ASM had the largest impact, with Accuracy dropping over 3% and LCS to 0.87. Omitting SNCC degraded FRA and robustness. These results demonstrate that all components contribute uniquely, jointly enabling MetaMath-LLaMA’s state-of-the-art performance.

8. Conclusion

In this paper, we proposed MetaMath-LLaMA, an advanced model for mathematical reasoning in large language models, introducing a meta-cognitive scheduler and a hybrid symbolic-neural computation cell. Our experimental results demonstrate that MetaMath-LLaMA significantly outperforms baseline models like GPT-3.5 across multiple evaluation metrics. Furthermore, the ablation results confirm the indispensable role of each core module—particularly the Answer Synthesizer and Meta-Cognitive Scheduler—in sustaining both accuracy and logical consistency. MetaMath-LLaMA sets a new standard for interpretable, consistent, and accurate mathematical reasoning in AI systems.

References

1. Pan, L.; Albalak, A.; Wang, X.; Wang, W.Y. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295* **2023**.
2. Yu, Y.; Zhang, Y.; Zhang, D.; Liang, X.; Zhang, H.; Zhang, X.; Yang, Z.; Khademi, M.; Awadalla, H.; Wang, J.; et al. Chain-of-Reasoning: Towards Unified Mathematical Reasoning in Large Language Models via a Multi-Paradigm Perspective. *arXiv preprint arXiv:2501.11110* **2025**.
3. Hu, Y.; Yu, Y. Enhancing neural mathematical reasoning by abductive combination with symbolic library. *arXiv preprint arXiv:2203.14487* **2022**.

4. Uesato, J.; Kushman, N.; Kumar, R.; Song, F.; Siegel, N.; Wang, L.; Creswell, A.; Irving, G.; Higgins, I. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275* **2022**.
5. Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; Cobbe, K. Let's verify step by step. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
6. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegrefe, S.; Alon, U.; Dziri, N.; Prabhunoye, S.; Yang, Y.; et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems* **2023**, *36*, 46534–46594.
7. Trinh, T.; Luong, T. AlphaGeometry: An Olympiad-level AI system for geometry. *Google DeepMind* **2024**, *17*.
8. Bandyopadhyay, D.; Bhattacharjee, S.; Ekbal, A. Thinking machines: A survey of llm based reasoning strategies. *arXiv preprint arXiv:2503.10814* **2025**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.