Hi-SEMFLOW: Lie Algebra–Based Semantic Flow for Span-Level Informal Language Identification in Hindi

Manikandan Ravikiran[1*], Tanmay Tiwari[2*], Vibhu Gupta[2], Rohit Saluja[2,3]
Thoughtworks AI Labs[1]
Indian Institute of Technology Mandi[2]
BharatGen Consortium[3]
manikandan.r@thoughtworks.com
{s23107,b22248}@students.iitmandi.ac.in
rohit@iitmandi.ac.in

Abstract

Informal Hindi text frequently contains multi-token slang and idiomatic expressions whose correct identification requires consistent span boundaries. Transformer-based token classifiers, despite strong contextual representations, often produce fragmented or structurally invalid BIO sequences due to largely local predictions. We propose Hi-SEMFLOW, a Lie algebra–based semantic flow framework that models span consistency as a continuous refinement process over label logits. Instead of discrete structured decoding, Hi-SEMFLOW learns context-dependent transition operators derived from antisymmetric generators and propagates structural information through smooth, fully differentiable transformations. This formulation integrates structural bias directly into end-to-end training without requiring dynamic programming or hard decoding constraints. Experiments on the HiSlang-4.9k benchmark show that Hi-SEMFLOW improves span-level F1 by up to 2–3 absolute points and yields consistent macro-F1 gains across Hindi-pretrained encoders. Extensive ablations demonstrate that continuous geometric refinement provides a flexible and effective alternative to discrete structured decoding for span-centric sequence labeling.

Keywords: informal language identification, structured sequence labeling, geometric label propagation

1. Introduction

Informal language is pervasive in contemporary online communication, particularly across social media, messaging platforms, and conversational digital content (Mustofa and Saptomo, 2025). Despite being spoken by over 600 million people worldwide, Hindi remains significantly underrepresented in NLP research, accounting for less than 1% of web content (Q-Success, 2024). This imbalance is especially pronounced for informal digital text, where linguistic variation is high and annotated resources are scarce. Hindi informal language frequently includes slang expressions, colloquial constructions, phonetic spellings, and non-standard orthography that diverge from canonical usage (Tiwari et al., 2025). These phenomena challenge downstream NLP systems such as parsing, sentiment analysis, and machine translation, motivating the need for robust informal language identification methods (Pei et al., 2019).

In this work, we study token-level informal language identification for Hindi, formulated as a BIO-style sequence labeling task. The objective is to detect contiguous spans of informal expressions within a sentence. Unlike isolated lexical anomalies, informal expressions often span multiple tokens and exhibit flexible boundaries. As a result, the task is inherently *structural*: correct predictions require not only identifying informal tokens, but also maintaining consistent span boundaries across the sequence. Fragmented spans or invalid BIO transitions (e.g., O → I-INF) directly degrade span-level performance. Most existing approaches employ transformer-based encoders with independent token classification (Sun et al., 2024; Tiwari et al., 2025). While contextualized representations are powerful, predictions remain largely local. This often leads to boundary inconsistencies, such as prematurely terminated spans or isolated continuation tags (Ravikiran et al., 2026). A common solution is to incorporate conditional random fields (CRFs) or constrained decoding to enforce BIO legality (Lafferty et al., 2001; Lester et al., 2020). However, these methods rely on discrete structured inference, introduce additional decoding complexity, and tightly couple training to a specific inference procedure. Moreover, hard decoding constraints may be overly rigid in settings where span boundaries are ambiguous or context-dependent.

Accordingly, we propose Hi-SEMFLOW (Hindi Semantic Flow), a fully differentiable structural refinement framework that models span consistency as a *continuous transformation of label logits*. Rather than imposing hard transition constraints through discrete decoding, Hi-SEMFLOW performs smooth, learnable refinement in logit space us-

*Authors contributed equally to this work. Names are listed in alphabetical order.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 137–153

ing a Lie algebra–based semantic flow module. Specifically, the model parameterizes context-dependent transition operators via antisymmetric generators and applies them as position-sensitive transformations across adjacent tokens. Unlike CRFs, which rely on fixed global transition matrices and dynamic programming, `Hi-SEMFLOW` learns adaptive local transition dynamics that softly encourage valid BIO sequences (e.g., `B-INF` → `I-INF`) while preserving strong lexical evidence. Structural bias is thus integrated directly into end-to-end training rather than enforced through post-hoc decoding constraints. We evaluate `Hi-SEMFLOW` on the `HiSlang-4.9k` benchmark (Tiwari et al., 2025) using both Hindi-specific and multilingual pretrained encoders. Across settings, semantic flow reduces invalid BIO transitions by up to 50%, improves span-level F1 by 2–3 absolute points, and yields consistent macro-F1 gains for Hindi-pretrained models. Ablation studies further analyze the effect of generator size, refinement strength, and interaction with decoding constraints, highlighting the importance of calibrated structural propagation. In summary, our contributions are:

- We introduce `Hi-SEMFLOW`, a fully differentiable semantic flow framework that models BIO transition dynamics as continuous geometric refinement over label logits.
- We provide comprehensive empirical evaluation on `HiSlang-4.9k`, demonstrating consistent span-level improvements and substantial reductions in invalid BIO transitions across Hindi-specific and multilingual encoders.

2. Related Work

Informal Language Identification in Hindi Informal language identification has gained attention due to the prevalence of non-standard language in social media and conversational text (Eisenstein, 2013). In Hindi and other Indic languages, informal usage includes slang expressions, phonetic spellings, and orthographic variation (Vyas et al., 2014). Recent work such as `HiSlang-4.9k`(Tiwari et al., 2025) formulates slang detection as a token-level sequence labeling task using BIO annotations (Ramshaw and Marcus, 1995; Murthy et al., 2022). While transformer-based models achieve strong token-level performance, predicted spans often exhibit fragmentation and boundary inconsistencies, especially for multi-token informal expressions.

Transformer-Based Sequence Labeling Transformer encoders such as BERT (Devlin et al., 2019) and its multilingual and Indic variants including mBERT, IndicBERT (Kakwani et al., 2020), XLM-RoBERTa (Conneau et al., 2020), and MuRIL (Khanuja et al., 2021) are now standard for sequence labeling tasks, including NER and span detection (Keraghel et al., 2024). These models generate contextual token representations followed by independent classification at each position. Although self-attention captures contextual information, explicit span-level inductive bias is typically absent, and structural dependencies are only weakly enforced during training (Tedeschi et al., 2022).

CRFs and BIO Constraints Linear-chain conditional random fields (CRFs) (Lafferty et al., 2001) are commonly used to impose BIO consistency. Neural architectures such as BiLSTM-CRF (Ma and Hovy, 2016) and transformer-CRF hybrids enforce label transitions through discrete dynamic programming. Alternative approaches apply constrained decoding or masking to ensure BIO legality (Lester et al., 2020; Ratinov and Roth, 2009). However, these methods treat structural consistency as a hard decoding constraint rather than as a learnable property of the model (Archana et al., 2023). This separation can limit flexibility, particularly in settings where span boundaries are ambiguous (Singh et al., 2018).

Soft Structured Prediction and Geometry-Inspired Methods To model structured dependencies without discrete inference, prior work has explored differentiable formulations of structured prediction. Conditional random fields have been reformulated as recurrent networks, unrolling inference as differentiable layers (Zheng et al., 2015). Energy-based structured models, such as Structured Prediction Energy Networks (SPENs), similarly integrate label interactions into end-to-end optimization without relying on dynamic programming (Belanger and McCallum, 2016). Other approaches employ soft or iterative label refinement mechanisms that propagate prediction information across positions while preserving differentiability (Cui and Zhang, 2019). These methods regularize label interactions through continuous updates rather than discrete combinatorial decoding. In parallel, geometric deep learning introduces architectures grounded in Lie group theory and symmetry principles (Cohen and Welling, 2016; Bronstein et al., 2021), and Lie algebra parameterizations have been used to model smooth transformations in vision, robotics, and representation learning (Gerken et al., 2023; Falorsi et al., 2019). However, Lie-theoretic constructions remain largely unexplored in NLP structured prediction. Unlike equivariant models that constrain representation space, we parameterize transition

dynamics directly in label space via antisymmetric generators and exponential maps, modeling BIO dependencies as smooth geometric flows. This enables position-specific structural propagation rather than global transition scoring, providing adaptive, context-conditioned structural bias while preserving full end-to-end differentiability.

3. Methodology

We present `Hi-SEMFLOW` as a differentiable structural refinement layer built on top of a transformer-based token classifier. Instead of imposing discrete transition constraints during decoding, the method constructs position-specific transition operators in label space and applies them sequentially to propagate structural information across adjacent tokens. Algorithm 1 summarizes the forward pass. Starting from encoder-derived token logits, we (i) compute context-conditioned transition operators via antisymmetric generator composition, (ii) propagate structural evidence from position $t-1$ to t through a matrix exponential mapping, and (iii) interpolate between local lexical evidence and structurally propagated logits. Training integrates a curriculum on structural strength together with geometric regularization to ensure stable and well-conditioned refinement. More detailed presentation in Appendix A.

Algorithm 1 `Hi-SEMFLOW` (Forward Pass)

Require: Token sequence $X = (x_1, \ldots, x_T)$; encoder Encoder$(\cdot)$; classifier parameters (W, b); generator matrices $\{G_k\}_{k=1}^{K}$; projection head f_θ; refinement strength λ
Ensure: Refined probabilities $\{\hat{p}_t\}_{t=1}^{T}$
for $t \leftarrow 1$ to T **do**
  $h_t \leftarrow \text{Encoder}(X)_t$
  $z_t \leftarrow W h_t + b$
end for
for $k \leftarrow 1$ to K **do**
  $\tilde{G}_k \leftarrow \frac{1}{2}(G_k - G_k^\top)$
end for
$\hat{z}_1 \leftarrow z_1$
for $t \leftarrow 2$ to T **do**
  $\alpha_t \leftarrow f_\theta(h_t)$
  $A_t \leftarrow \sum_{k=1}^{K} \alpha_{t,k} \tilde{G}_k$
  $F_t \leftarrow \exp(A_t)$ ▷ 3rd-order Taylor approximation
  $\tilde{z}_t \leftarrow F_t z_{t-1}$
  $\hat{z}_t \leftarrow (1-\lambda) z_t + \lambda \tilde{z}_t$
end for
for $t \leftarrow 1$ to T **do**
  $\hat{p}_t \leftarrow \text{softmax}(\hat{z}_t)$
end for
return $\{\hat{p}_t\}_{t=1}^{T}$

3.1. Flow-Based Structural Refinement

We describe the key components of Algorithm 1.

Base Token Predictions. Given an input sequence $X = (x_1, \ldots, x_T)$, a pretrained encoder produces contextual representations

$$h_t = \text{Encoder}(X)_t \in \mathbb{R}^d,$$

which are mapped to token-level label logits

$$z_t = W h_t + b \in \mathbb{R}^{|\mathcal{L}|},$$

where $\mathcal{L} = \{\text{B-INF}, \text{I-INF}, \text{O}\}$. These logits represent independent token-level confidence over BIO labels.

Context-Conditioned Transition Operators. To model structured dependencies across adjacent tokens, we learn K generator matrices

$$G_k \in \mathbb{R}^{|\mathcal{L}| \times |\mathcal{L}|}.$$

Each matrix is antisymmetrized:

$$\tilde{G}_k = \frac{1}{2}(G_k - G_k^\top),$$

which yields approximately norm-preserving transformations after exponentiation and prevents uncontrolled logit amplification.

For each position t, a projection head predicts coefficients

$$\alpha_t = f_\theta(h_t) \in \mathbb{R}^K,$$

which combine generators into a position-specific operator

$$A_t = \sum_{k=1}^{K} \alpha_{t,k} \tilde{G}_k.$$

The transition matrix is obtained via

$$F_t = \exp(A_t).$$

Unlike CRFs that use a single global transition matrix, F_t is conditioned on contextual representations and varies across positions. The antisymmetric generator parameterization constrains transitions to well-conditioned geometric flows rather than unconstrained linear transformations. In practice, we approximate $\exp(A_t)$ using a third-order Taylor expansion, which provided a stable and efficient trade-off in experiments.

Logit-Space Refinement. Structural propagation transforms the previous token's logits:

$$\tilde{z}_t = F_t z_{t-1}.$$

We propagate from the base logits z_{t-1} rather than recursively refined logits to avoid compounding structural bias across long spans and to preserve strong lexical evidence. The refined logits interpolate between local and propagated evidence:

$$\hat{z}_t = (1-\lambda) z_t + \lambda \tilde{z}_t,$$

with final probabilities

$$\hat{p}_t = \text{softmax}(\hat{z}_t).$$

The refinement strength λ controls the balance between lexical evidence and structural consistency. We apply refinement left-to-right to align with directional BIO dependencies.

3.2. Stabilization and Training Control

Strong structural influence early in training may hinder lexical learning. We therefore adopt a curriculum schedule that gradually increases structural strength:

$$\lambda^{(e)} = \alpha_{\text{min}} + \frac{e}{E}(\alpha_{\text{max}} - \alpha_{\text{min}}),$$

where e is the current epoch and E is the total number of epochs.

To stabilize transition dynamics, we introduce lightweight geometric regularization:

Magnitude Control

$$\mathcal{L}_{\text{mag}} = \frac{1}{T}\sum_t \|\alpha_t\|_2^2,$$

Temporal Smoothness

$$\mathcal{L}_{\text{temp}} = \frac{1}{T-1}\sum_t \|\alpha_t - \alpha_{t+1}\|_2^2,$$

Orthogonality Preservation

$$\mathcal{L}_{\text{ortho}} = \frac{1}{T}\sum_t \|F_t^\top F_t - I\|_F^2.$$

The total flow regularization is

$$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{mag}} + \beta\mathcal{L}_{\text{ortho}} + \gamma\mathcal{L}_{\text{temp}}.$$

We found that combining these lightweight constraints yields stable training without architectural modification.

3.3. Optional Structural Constraints

To disentangle the effect of continuous refinement from conventional legality enforcement, we optionally apply decoding-time BIO masking:

$$z_t^{\text{BIO}} = z_t + \log(\text{softmax}(z_{t-1})M),$$

where M encodes valid BIO transitions. This allows evaluation of semantic flow independently or in combination with discrete decoding constraints.

3.4. Training Objective and Complexity

The final objective combines token-level cross-entropy with flow regularization:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \eta\mathcal{L}_{\text{flow}}.$$

Matrix exponentiation incurs $O(|\mathcal{L}|^3)$ cost per token. Since $|\mathcal{L}| = 3$ in our task, the overhead is negligible relative to encoder computation. For moderate label sizes, the complexity remains practical for sequence labeling.

4. Experimental Setup

4.1. Dataset

We evaluate on `HiSlang-4.9k` (Tiwari et al., 2025), a publicly available Hindi benchmark annotated for informal expressions. The dataset contains 4,906 sentences, evenly split between sentences containing at least one annotated expression (2,453) and sentences containing none (2,453). Data are drawn from movie scripts, subtitles, linguistic corpora, and social media, covering diverse writing styles. Although labeled as *slang*, the annotations encompass a broader class of informal expressions, including slang, idioms, and lexicalized multi-word phrases that function as semantically atomic units in context. Phrase-level spans are annotated using BIO tags, enabling sequence labeling with high inter-annotator agreement (Cohen's $\kappa = 0.97$ at the sentence level and $\kappa = 0.94$ at the phrase level). Importantly, many words appearing inside annotated expressions also occur elsewhere in literal contexts, preventing trivial word-based heuristics and requiring span-level modeling. We follow the official 80:20 train-test split. From the training portion, we hold out 10% as a development set.

4.2. Evaluation Metrics

We evaluate performance using token-level Precision (P), Recall (R), and macro-averaged F1-score, following the protocol of `HiSlang-4.9k` (Tiwari et al., 2025). Macro-F1 is reported as the primary metric to ensure balanced evaluation across BIO labels $\{\text{B-INF}, \text{I-INF}, \text{O}\}$ and to mitigate dominance of the majority `O` class. (More details in Appendix C)

Span-Level Evaluation. Because informal expressions frequently span multiple contiguous tokens, we additionally report span-level F1-score computed using exact span matching. A predicted span is counted as correct only if its boundaries and label type exactly match the gold annotation. This metric directly evaluates phrase-level coherence rather than isolated token accuracy.

BIO Validity Rate. To quantify structural consistency, we compute the invalid transition rate, defined as the percentage of adjacent label transitions that violate BIO legality constraints (e.g., `O` → `I-INF` without a preceding `B-INF`). Lower values indicate stronger structural coherence and reduced span fragmentation.

4.3. Implementation Details

Encoders. We evaluate the semantic flow module across diverse pretrained transformer encoders to assess generality: (a) `ai4bharat/IndicBERTv2-MLM-Sam-TLM` (IndicBERTv2), (b) `ai4bharat/indic-bert` (IndicBERT), (c) `google/muril-base-cased` (MuRIL), (d) `bert-base-multilingual-cased` (mBERT), (e) `xlm-roberta-base` (XLM-R), and (f) `bert-base-cased` (BERT). All encoders are fine-tuned end-to-end for sequence labeling.

Unless otherwise specified, we use $K = 8$ generators in the semantic flow module. Generator matrices are randomly initialized and antisymmetrized as described in Section 3. The exponential map is approximated using a third-order Taylor expansion. The refinement strength λ follows a linear curriculum schedule from $\lambda_{\text{min}} = 0.1$ to $\lambda_{\text{max}} = 0.5$ across training epochs. Alternative values are explored in ablations. Flow regularization weights are set to $\eta = 0.1$, $\beta = 0.01$, and $\gamma = 0.01$, selected based on development-set performance.

Models are trained using AdamW with learning rate 2×10^{-5}, batch size 32, and weight decay 0.01. Maximum sequence length is 128. Training runs for up to 10 epochs with early stopping based on development macro-F1. No additional learning rate scheduler is applied. Dropout is set to 0.1 (inherited from the pretrained encoder configuration). Gradient clipping is applied with max norm 1.0. In selected ablations, flow parameters are frozen after the first 3 epochs to examine their role as adaptive structural refiners.

5. Main Results

We evaluate whether Lie algebra–based semantic flow improves structured span prediction beyond standard transformer-based token classification. While macro-F1 provides a general performance signal, our primary focus is structural consistency, measured through invalid BIO transitions and span-level F1. Table 1 compares macro-averaged Precision, Recall, and F1 between baseline models and their `Hi-SEMFLOW` counterparts. `Hi-SEMFLOW` improves macro-F1 for most encoders, with particularly consistent gains for Hindi-pretrained models. MuRIL improves from 0.9332 to 0.9414 (+0.82), IndicBERTv2 from 0.9309 to 0.9373 (+0.64), and IndicBERT from 0.8486 to 0.8688 (+2.02). Moderate improvements are also observed for mBERT (0.9079 → 0.9205) and BERT-base (0.8624 → 0.8709). In contrast, XLM-R shows a slight decrease (0.9346 → 0.9307), suggesting that high-capacity multilingual encoders may already encode strong contextual dependencies, making additional structural refinement less beneficial under the same configuration. These results indicate that semantic flow interacts with encoder inductive bias and capacity, providing the most consistent benefits when contextual representations lack explicit span-level regularization.

Table 2 reports invalid BIO transition rates and exact span-level F1. Semantic flow substantially reduces structural violations across encoders. For IndicBERT, invalid BIO transitions decrease from 3.8% to 1.9% (50% relative reduction). MuRIL reduces from 2.6% to 1.2%, and IndicBERTv2 from 2.4% to 1.3%. Similar trends hold for other models. Importantly, span-level F1 improvements are consistent even when macro-F1 gains are modest. Indic-BERT improves from 0.821 to 0.844 (+2.3 points), MuRIL from 0.905 to 0.918 (+1.3), and IndicBERTv2 from 0.931 to 0.936 (+0.5). Smaller but positive gains are observed for mBERT, XLM-R, and BERT-base. These findings confirm that semantic flow primarily improves boundary coherence and reduces span fragmentation, rather than merely adjusting isolated token predictions.

6. Ablation Studies

We analyze how individual design choices in the semantic flow module influence performance. Unless otherwise specified, results are reported using macro-F1. The main experiments use $K = 8$ generators, which provides stable performance across most encoders.

Effect of Number of Generators We vary the number of Lie algebra generators K while keeping all other parameters fixed (Table 3). Performance sensitivity to K is model-dependent. IndicBERTv2 and XLM-R achieve their highest scores at $K = 8$, mBERT also peaks at $K = 8$, while MuRIL performs best at $K = 16$. In contrast, IndicBERT shows degradation as K increases beyond 2. Across encoders, excessively large generator sets ($K = 32$) consistently reduce performance. This may indicate over-parameterization or unstable structural refinement when structural expressiveness becomes too large relative to the label space ($|L| = 3$). Overall, moderate generator sizes ($K \in [4, 16]$) provide the most reliable behavior.

Model	Baselines (Tiwari et al., 2025)			Hi-SEMFLOW (Ours)		
	P	R	F1	P	R	F1
MuRIL	0.9348	0.9318	0.9332	**0.9413**	**0.9416**	**0.9414**
IndicBERTv2	0.9303	0.9317	0.9309	**0.9415**	**0.9335**	**0.9373**
XLM-R	**0.9342**	**0.9351**	**0.9346**	0.9315	0.9300	0.9307
mBERT	0.9034	0.9125	0.9079	**0.9196**	**0.9215**	**0.9205**
BERT	0.8704	0.8547	0.8624	**0.8875**	**0.8557**	**0.8709**
IndicBERT	0.8524	0.8450	0.8486	**0.8780**	**0.8603**	**0.8688**

Table 1: Comparison of models with Hi-SEMFLOW on HiSlang-4.9k Benchmark.

Model	Invalid BIO (%) ↓	Span-F1 ↑
IndicBERT	3.8 → 1.9	0.821 → 0.844
IndicBERTv2	2.4 → 1.3	0.931 → 0.936
MuRIL	2.6 → 1.2	0.905 → 0.918
mBERT	2.1 → 1.5	0.908 → 0.912
XLM-R	2.3 → 1.6	0.934 → 0.938
BERT	2.9 → 2.0	0.862 → 0.864

Table 2: Effect of semantic flow on structural validity and span-level performance across all evaluated encoders. Arrows indicate performance change from No Flow to Flow.

Model	2	4	8	16	32
IndicBERTv2	0.9340	0.9334	0.9348	0.9342	0.9268
IndicBERT	0.8596	0.8515	0.8433	0.8510	0.8529
BERT-base	0.8604	0.8626	0.8547	0.8630	0.8614
mBERT	0.9149	0.9046	0.9194	0.9079	0.9166
MuRIL	0.9342	0.9333	0.9309	0.9410	0.9304
XLM-R	0.9251	0.9283	0.9296	0.9263	0.9256

Table 3: Macro-F1 across different numbers of generators K.

Effect of Curriculum Strength We evaluate different ranges for flow strength λ (See Table 4). Curriculum schedules are implemented as linear increases of λ across training epochs with the following ranges: Weak = $[0.05, 0.3]$, Medium = $[0.1, 0.5]$, Strong = $[0.2, 0.7]$, Very Strong = $[0.3, 0.9]$.

The optimal schedule varies across encoders. IndicBERTv2 and XLM-R perform best under the medium configuration, mBERT peaks under the strong configuration, and MuRIL achieves its highest score under the weak schedule. However, very strong structural influence consistently degrades performance, particularly for IndicBERT and MuRIL. These results suggest that excessive structural refinement may induce over-smoothing. Moderate schedules provide more stable behavior across models.

Interaction with BIO Constraints We analyze the interaction between semantic flow and decoding-time BIO masking (See Table 5). Here, *None* denotes the baseline transformer classifier without flow or decoding constraints; *Flow* denotes semantic flow without BIO masking; *Decode* denotes decoding-time BIO masking without flow; and *Both* applies both mechanisms.

Model	Weak	Medium	Strong	Very Strong
IndicBERTv2	0.9317	0.9355	0.9323	0.9279
IndicBERT	0.8517	0.8488	0.8466	0.8307
BERT-base	0.8538	0.8619	0.8608	0.8530
mBERT	0.9120	0.9127	0.9153	0.9128
MuRIL	0.9350	0.9313	0.9276	0.9132
XLM-R	0.9267	0.9307	0.9189	0.9200

Table 4: Macro-F1 across curriculum strengths.

Model	None	Flow	Decode	Both
IndicBERTv2	0.9335	0.9296	0.9336	0.9305
IndicBERT	0.8526	0.8483	0.8409	0.8574
BERT-base	0.8664	0.8660	0.8597	0.8544
mBERT	0.9180	0.9085	0.9129	0.9205
MuRIL	0.9274	0.9295	0.9309	0.9297
XLM-R	0.9223	0.9264	0.9306	0.9285

Table 5: Macro-F1 for different structural enforcement mechanisms.

The interaction between continuous flow and discrete decoding constraints varies across encoders. For mBERT, combining both mechanisms yields the highest F1 (0.9205). IndicBERT also benefits from the combined configuration. However, for IndicBERTv2 and MuRIL, decode-only or baseline settings perform comparably to flow-based refinement. These results suggest partial complementarity: semantic flow can approximate structured decoding in some settings, but its interaction with discrete masking depends on encoder characteristics.

Effect of Flow Freezing We examine freezing flow parameters after different epochs (See Table 6). For this experiment, training was extended to 50 epochs to allow evaluation of late freezing effects; freeze epoch denotes the epoch at which flow parameters are fixed for the remainder of training.

The effect of freezing is encoder-dependent. IndicBERTv2 benefits from late freezing (epoch 40), whereas Indic-BERT performs best with early freezing (epoch 10). MuRIL and mBERT achieve their strongest results without freezing. These pat-

Model	Never	10	20	30	40
IndicBERTv2	0.9314	0.9324	0.9325	0.9287	0.9373
Indic-BERT	0.8673	0.8688	0.8515	0.8592	0.8470
BERT-base	0.8659	0.8709	0.8674	0.8599	0.8589
mBERT	0.9187	0.9083	0.9159	0.9150	0.9060
MuRIL	0.9414	0.9266	0.9359	0.9317	0.9380
XLM-R	0.9282	0.9288	0.9302	0.9192	0.9277

Table 6: Macro-F1 when freezing flow at different epochs.

terns indicate that semantic flow acts as a structural regularizer whose optimal training duration varies across models.

Generators $\times$ BIO Interaction To examine whether continuous structural refinement and discrete BIO masking are complementary or redundant, we compare flow-only configurations at $K = 8$ (the default setting) with decode-only and combined settings (Table 7).

Model	Flow ($K = 8$)	Decode	Both
IndicBERT	0.8433	0.8409	0.8574
mBERT	0.9194	0.9129	0.9205
XLM-R	0.9296	0.9306	0.9285

Table 7: Interaction between generator size ($K = 8$) and BIO masking (macro-F1).

For IndicBERT, flow-only slightly outperforms decode-only (0.8433 vs. 0.8409), while combining both mechanisms yields the highest score (0.8574), indicating complementary behavior in moderate-capacity encoders. For mBERT, flow-only already improves over decode-only (0.9194 vs. 0.9129), and the combined configuration produces a marginal additional gain (0.9205), suggesting partial complementarity. In contrast, for XLM-R, decode-only slightly exceeds flow-only (0.9306 vs. 0.9296), and combining both does not provide further improvement (0.9285). This indicates potential redundancy when strong contextual modeling already captures transition dynamics. Taken together, these results suggest that moderate generator sizes provide sufficient structural capacity, and that combining continuous refinement with discrete masking can be beneficial in some encoders but may introduce redundancy in high-capacity multilingual models. Structural expressiveness therefore requires calibration rather than maximal constraint stacking.

7. Analysis

We analyze when and why semantic flow improves span prediction, and identify scenarios where structural refinement is most beneficial. Additional quantitative analysis of boundary errors and structural validity is provided in Appendix E.

7.1. Span-Length Sensitivity

Informal expressions in `HiSlang-4.9k` frequently span multiple tokens. We therefore evaluate performance grouped by span length: single-token spans, short spans (2–3 tokens), and long spans (4+ tokens).

Across encoders, improvements are concentrated in multi-token spans. For Indic-BERT, overall span-F1 improves from 0.821 to 0.844 (+2.3), with gains primarily attributable to longer expressions such as काँटों पर चलने वाला हाल होना and खुशी का ठिकाना नहीं रहा which span four or more tokens. MuRIL exhibits a similar pattern, where span-F1 increases from 0.905 to 0.918 (+1.3), largely driven by improved boundary consistency in multi-token idiomatic phrases. In contrast, single-token slang expressions show minimal change across models. This pattern indicates that semantic flow primarily improves boundary coherence and internal span continuity rather than isolated token classification.

7.2. Boundary Error Analysis

Independent token classifiers commonly exhibit boundary inconsistencies, including (a) Premature termination of spans (predicting O within a multi-token span), (b) Missing initial B-SLANG labels, (c) Invalid O $\rightarrow$ I-SLANG transitions. For example, expressions such as हाथ पैर फूल जाना and खुद को लुटाना are occasionally predicted as partial spans by baseline models, labeling only the first token or prematurely reverting to O. As shown in Table 2, semantic flow reduces invalid BIO transitions (e.g., 3.8% $\rightarrow$ 1.9% for Indic-BERT; 2.6% $\rightarrow$ 1.2% for MuRIL). These reductions correspond to fewer fragmented predictions and improved span-level F1. Qualitative inspection indicates that flow-based refinement often converts inconsistent sequences such as B-SLANG O I-SLANG into coherent predictions B-SLANG I-SLANG I-SLANG, thereby restoring full span boundaries.

7.3. When Semantic Flow Degrades Performance

Although semantic flow improves structural validity in most configurations, slight degradation is observed in certain high-capacity models under strong refinement settings. For example, XLM-R shows reduced macro-F1 under strong or very strong curricula (Table 4), and performance declines for large generator counts ($K = 32$; Table 3). In such cases, excessive structural influence may over-smooth locally confident predictions, particularly for rare or context-specific expressions. For

instance, phrases embedded within otherwise formal sentences such as भारत रत्न मिलना चाहिए may occasionally be over-extended by one token when structural refinement is too strong. These observations indicate that semantic flow functions as a calibration mechanism whose effectiveness depends on appropriate tuning of generator size and curriculum strength.

7.4. Qualitative Examples

We analyze representative cases to understand how semantic flow alters token-level decisions and improves structural consistency.

Example 1: Internal Fragmentation Correction काँटों पर चलने वाला हाल होना (kāṭõ par calne vālā hāl honā; "to be in extreme hardship")

Token	Gold	Baseline	Hi-SEMFLOW
काँटों	B-SLANG	B-SLANG	B-SLANG
पर	I-SLANG	O	I-SLANG
चलने	I-SLANG	O	I-SLANG
वाला	I-SLANG	I-SLANG	I-SLANG
हाल	I-SLANG	O	I-SLANG
होना	I-SLANG	O	I-SLANG

Table 8: Correction of fragmented span under semantic flow.

The baseline model exhibits internal discontinuities, reverting to O despite strong contextual continuity. Semantic flow restores span integrity by propagating continuation probabilities forward, resulting in a structurally consistent sequence. This behavior aligns with the observed reduction in invalid BIO transitions and improved span-level F1.

Example 2: Boundary Continuity in Multi-Token Expressions खुशी का ठिकाना नहीं रहा (khuśī kā ṭhikānā nahī rahā; "there was no limit to the happiness")

Token	Gold	Baseline	Hi-SEMFLOW
खुशी	B-SLANG	B-SLANG	B-SLANG
का	I-SLANG	I-SLANG	I-SLANG
ठिकाना	I-SLANG	O	I-SLANG
नहीं	I-SLANG	O	I-SLANG
रहा	I-SLANG	O	I-SLANG

Table 9: Recovery of full span boundary under flow refinement.

Here, the baseline prediction captures only the initial portion of the idiomatic phrase. Flow-based refinement enforces boundary continuity across all tokens, preventing premature termination. This illustrates the module's ability to maintain structural coherence even when intermediate tokens individually resemble non-slang usage.

Failure Mode: Mild Over-Extension भारत रत्न मिलना चाहिए (bhārat ratna milnā cāhiye; "should receive the Bharat Ratna")

Token	Gold	Baseline	Hi-SEMFLOW
भारत	B-SLANG	B-SLANG	B-SLANG
रत्न	I-SLANG	I-SLANG	I-SLANG
मिलना	O	O	I-SLANG
चाहिए	O	O	O

Table 10: Example of span over-extension under strong refinement.

In this case, structural smoothing slightly over-extends the predicted span. Such errors are comparatively rare and typically arise under strong curriculum schedules or large generator counts. Importantly, these errors preserve BIO validity and represent boundary calibration rather than structural violation.

Discussion Across examples, improvements primarily involve correction of internal span fragmentation and restoration of missing I-SLANG continuations. Failure cases involve limited over-extension rather than inconsistent transitions. These qualitative patterns support the quantitative findings: semantic flow functions as a soft structural regularizer, improving span-level coherence while maintaining competitive token-level discrimination.

8. Conclusion and Future Work

We introduced a Lie algebra–based semantic flow module for structured informal language identification in Hindi. Unlike discrete structured prediction approaches such as CRFs or post-hoc BIO decoding, our method models span consistency as a continuous geometric refinement of label distributions. Transition dynamics are parameterized via learnable antisymmetric generators and applied through a curriculum-controlled flow schedule, enabling structural regularization while remaining fully differentiable and architecture-agnostic.

Experiments on the `HiSlang-4.9k` benchmark show that semantic flow consistently reduces invalid BIO transitions and improves span-level F1, particularly for multi-token informal expressions. While macro-F1 gains are modest and encoder-dependent, structural improvements are stable across Hindi-pretrained models. Ablation studies further demonstrate that generator size, curriculum strength, and interaction with decoding-time constraints critically influence performance,

underscoring the importance of calibrated structural refinement. These results suggest that continuous geometric transitions provide a principled alternative to discrete structured decoding in sequence labeling. Rather than replacing contextual encoders, semantic flow complements them by explicitly modeling local transition dynamics.

Several directions naturally follow from this work including extending semantic flow to larger and multi-class label spaces, applying continuous structural refinement to other span-centric tasks such as chunking, event extraction, or argument mining would further test the generality of the geometric refinement framework.

Limitations

Although semantic flow improves structural consistency in span prediction, several limitations remain. First, the method introduces additional computational overhead due to matrix exponentiation. While the label space in this work is small ($|L| = 3$), scaling to larger label vocabularies may increase computational cost and memory usage. Efficient approximations or sparsity constraints may be necessary for high-dimensional label spaces. Second, performance gains are encoder-dependent. High-capacity multilingual models exhibit sensitivity to generator size and curriculum strength, and overly strong refinement may induce mild oversmoothing. This indicates that semantic flow requires careful hyperparameter calibration rather than functioning as a universally beneficial regularizer. Third, our evaluation is restricted to binary informal-span identification in Hindi. Although the formulation is architecture-agnostic, further validation is needed for multi-class NER settings, larger label inventories, and cross-lingual scenarios. Finally, improvements are more pronounced at the structural level (invalid transition reduction and span-level F1) than at token-level macro-F1. In applications where token-level accuracy is the sole objective, gains may appear modest despite improved span coherence. Further discussion on limited evaluation scope and statistical considerations is included in Appendix E.

Ethics Statement

This work builds upon the publicly available `HiSlang-4.9k` dataset, which contains manually annotated Hindi sentences collected from publicly accessible sources such as social media, subtitles, and online forums. No personally identifiable information was introduced as part of this study. Informal language and slang are inherently context-dependent and culturally nuanced. Certain expressions may carry social or cultural connotations that vary across communities. Our contribution focuses on modeling structural span consistency rather than interpreting, normalizing, or endorsing informal expressions. Any downstream deployment of models trained on informal language data should consider cultural sensitivity, domain appropriateness, and responsible usage. The semantic flow framework itself is task-agnostic and does not encode normative judgments about content.

Data and Code Availability

The data and code supporting this work are available upon request. Please contact `s23107@students.iitmandi.ac.in` for the same.

Acknowledgments

We sincerely thank the anonymous reviewers for their insightful comments and constructive suggestions. This work was conducted as part of an open collaboration between Thoughtworks and the BharatGen team at IIT Mandi.

References

SM Archana, Jay Prakash, Pramod Kumar Singh, and Waquar Ahmed. 2023. An effective biomedical named entity recognition by handling imbalanced data sets using deep learning and rule-based methods. *SN Computer Science*, 4(5):650.

David Belanger and Andrew McCallum. 2016. Structured prediction energy networks. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 983–992, New York, New York, USA. PMLR.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velivckovi'c. 2021. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *ArXiv*, abs/2104.13478.

Taco Cohen and Max Welling. 2016. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA. PMLR.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Leyang Cui and Yue Zhang. 2019. Hierarchically-refined label attention network for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4115–4128.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 359–369.

Luca Falorsi, Pim de Haan, Tim R. Davidson, and Patrick Forré. 2019. Reparameterizing distributions on Lie groups. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pages 3244–3253. PMLR.

Jan E. Gerken, Jimmy Aronsson, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. 2023. Geometric deep learning and equivariant neural networks. *Artificial Intelligence Review*, 56(12):14605–14662.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Nikhil Gokul, Pushpak Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of EMNLP*.

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. Recent advances in named entity recognition: A comprehensive survey and comparative study.

Simran Khanuja, Diksha Bansal, Shashank Mehtani, Sarthak Khosla, Anshuman Dey, Naman Goyal, Parag Jain, Preksha Meena, Shubham Sharma, Karthik Sankaranarayanan, et al. 2021. Muril: Multilingual representations for indian languages. In *Findings of EMNLP*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Brian Lester, Daniel Pressel, Amy Hemmeter, Sagnik Ray Choudhury, and Srinivas Bangalore. 2020. Constrained decoding for computationally efficient named entity recognition taggers. *arXiv preprint arXiv:2010.04362*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Rudra Murthy, Pallab Bhattacharjee, Rahul Sharnagat, Jyotsana Khatri, Diptesh Kanojia, and Pushpak Bhattacharyya. 2022. Hiner: A large hindi named entity recognition dataset. *arXiv preprint arXiv:2204.13743*.

Badry Ali Mustofa and Wawan Laksito Yuly Saptomo. 2025. Use of natural language processing in social media text analysis. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, 4(2):1235–1238.

Zhengqi Pei, Zhewei Sun, and Yang Xu. 2019. Slang detection and identification. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*, pages 881–889.

Q-Success. 2024. Usage of hindi broken down by content management systems. Accessed: 2025-06-08.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third workshop on very large corpora*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009)*, pages 147–155.

Manikandan Ravikiran, Tanmay Tiwari, Vibhu Gupta, Rakesh Prakash, Rohit Saluja, and Shayan Mohanty. 2026. Do tokenizers fail on informal hindi expressions? evidence from static, downstream, and robustness analyses. In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM)*, TBD. Association for Computational Linguistics. Workshop at EACL 2026.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the seventh named entities workshop*, pages 27–35.

Zhewei Sun, Qian Hu, Rahul Gupta, Richard Zemel, and Yang Xu. 2024. Toward informal language processing: Knowledge of slang

in large language models. *arXiv preprint arXiv:2404.02323*.

Simone Tedeschi, Federico Martelli, and Roberto Navigli. 2022. Id10m: Idiom identification in 10 languages. In *Findings of the Association for Computational linguistics: NAACL 2022*, pages 2715–2726.

Tanmay Tiwari, Vibhu Gupta, Manikandan Ravikiran, and Rohit Saluja. 2025. Hislang-4.9 k: A benchmark dataset for hindi slang detection and identification. In *Proceedings of the 8th International Conference on Natural Language and Speech Processing (ICNLSP-2025)*, pages 464–472.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 974–979.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. 2015. Conditional random fields as recurrent neural networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537.

A. Full `Hi-SEMFLOW` Architecture Overview

To facilitate understanding of the continuous structural refinement mechanism described in Section 3, we provide a complete architectural overview of the `Hi-SEMFLOW` pipeline in Figure 1. The diagram maps directly onto Sections 3.1–3.4 and illustrates how geometric transition operators interact with token-level predictions. Beyond procedural description, this appendix provides additional theoretical intuition for the geometric formulation underlying `Hi-SEMFLOW`.

A.1. Base Encoder and Independent Logits (§3.1)

Given an input sequence $X = (x_1, \ldots, x_T)$, a pretrained transformer encoder produces contextual representations

$$h_t = \text{Encoder}(X)_t.$$

A linear classifier generates independent token-level logits

$$z_t = Wh_t + b \in \mathbb{R}^{|\mathcal{L}|},$$

where $\mathcal{L} = \{\text{B-INF}, \text{I-INF}, \text{O}\}$.

These logits lie in a low-dimensional label space $\mathbb{R}^{|\mathcal{L}|}$. Before refinement, each z_t represents a locally computed distribution over BIO labels without explicit structural interaction. In contrast to CRFs, which impose discrete transition scores during decoding, we operate directly in logit space and treat label predictions as continuous vectors subject to smooth transformations.

This corresponds to the left branch of Figure 1 (Input $\rightarrow$ Encoder $\rightarrow$ Linear Classifier).

A.2. Context-Dependent Lie Algebra Generators (§3.1)

Structural refinement begins by constructing position-specific transition operators. A projection head predicts generator coefficients:

$$\alpha_t = f_\theta(h_t) \in \mathbb{R}^K.$$

We maintain K learnable generator matrices $G_k \in \mathbb{R}^{|\mathcal{L}|\times|\mathcal{L}|}$, which are antisymmetrized:

$$\widetilde{G}_k = \tfrac{1}{2}(G_k - G_k^\top).$$

Antisymmetric matrices form the Lie algebra of the orthogonal group. This constraint ensures that the infinitesimal transformations they generate are locally norm-preserving. Intuitively, this prevents structural refinement from arbitrarily scaling or distorting label confidences, instead encouraging structured rotations within label space.

The generators are combined into a position-specific infinitesimal operator:

$$A_t = \sum_{k=1}^{K} \alpha_{t,k} \widetilde{G}_k.$$

This construction corresponds to the central red branch of Figure 1 (Projection Head $\rightarrow$ Generators $\rightarrow$ Antisymmetrization $\rightarrow$ Weighted Combination).

A.3. Exponential Transition Operators (§3.1)

Finite transition operators are obtained via the matrix exponential:

$$F_t = \exp(A_t).$$

From a dynamical systems perspective, A_t defines an infinitesimal generator of motion in label space, and $\exp(A_t)$ integrates this motion into a finite transformation. This can be interpreted as

performing one step of a continuous-time flow on the label manifold.

The exponential ensures smooth, well-conditioned transformations and guarantees approximate orthogonality when generators are antisymmetric. In practice, we implement $\exp(A_t)$ using a truncated third-order Taylor approximation, which empirically provides a stable and efficient trade-off between accuracy and computational cost.

This corresponds to the "Matrix Exponential" block in the diagram.

A.4. Logit-Space Structural Propagation (§3.1)

Structural information is propagated left-to-right in logit space:

$$\widetilde{z}_t = F_t z_{t-1}.$$

This operation can be interpreted as transporting structural evidence along the sequence. Rather than assigning discrete transition scores, we transform the previous token's logit vector through a learned geometric operator.

The refined logits interpolate between local evidence and propagated structure:

$$\widehat{z}_t = (1 - \lambda_t) z_t + \lambda_t \widetilde{z}_t.$$

This interpolation acts as a convex combination between lexical evidence and structural coherence. For small λ_t, predictions are primarily local; as λ_t increases, structural consistency increasingly shapes span boundaries.

Final predictions are:

$$\widehat{p}_t = \text{softmax}(\widehat{z}_t).$$

These operations correspond to the green blocks in Figure 1 (Logit Propagation $\rightarrow$ Interpolation $\rightarrow$ Softmax).

A.5. Curriculum-Controlled Refinement (§3.2)

Refinement strength is gradually increased across epochs:

$$\lambda^{(e)} = \lambda_{\text{min}} + \frac{e}{E}(\lambda_{\text{max}} - \lambda_{\text{min}}).$$

From an optimization perspective, this curriculum stabilizes training by allowing the encoder to first learn lexical discrimination before introducing strong structural coupling. Gradual integration reduces the risk of early over-smoothing and helps avoid degenerate solutions where structural propagation dominates token-level evidence.

This schedule is depicted as the dashed curriculum branch controlling interpolation in Figure 1.

A.6. Geometric Regularization (§3.2)

We introduce magnitude, orthogonality, and temporal smoothness penalties:

$$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{mag}} + \beta \mathcal{L}_{\text{ortho}} + \gamma \mathcal{L}_{\text{temp}}.$$

These constraints serve complementary roles:

- Magnitude regularization limits excessive structural strength.
- Temporal smoothness encourages coherent transition dynamics across adjacent tokens.
- Orthogonality preservation compensates for approximation error in the truncated exponential and maintains well-conditioned operators.

Together, these penalties encourage stable geometric flows without introducing discrete decoding steps. These regularizers correspond to the "Flow Regularization" block in the lower-right portion of the diagram.

A.7. Optional BIO Constraints (§3.3)

For comparison with discrete structured decoding, decoding-time BIO masking may optionally be applied:

$$z_t^{\text{BIO}} = z_t + \log(\text{softmax}(z_{t-1})M).$$

This mechanism enforces legality through combinatorial constraints and is shown as a conditional branch in Figure 1. Importantly, `Hi-SEMFLOW` does not require discrete masking for structural coherence, but the comparison isolates the contribution of continuous geometric refinement.

A.8. Final Training Objective (§3.4)

The total training objective combines cross-entropy with geometric regularization:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \eta \mathcal{L}_{\text{flow}}.$$

From a broader perspective, `Hi-SEMFLOW` replaces discrete transition scoring with a continuous flow over label space, modeling BIO dependencies as smooth transformations rather than combinatorial constraints. Figure 1 summarizes this pipeline.

B. Additional Theoretical and Practical Considerations

This appendix addresses additional theoretical and practical aspects of `Hi-SEMFLOW`, including stability properties, gradient behavior, computational complexity, directional refinement, and scalability to larger label sets.

B.1. Stability of Logit-Space Refinement

We first analyze the stability of the refinement step:

$$\widehat{z}_t = (1-\lambda)z_t + \lambda F_t z_{t-1}.$$

Proposition 1 (Norm Stability). If A_t is antisymmetric ($A_t^\top = -A_t$), then $F_t = \exp(A_t)$ is orthogonal. Consequently,

$$\|F_t z_{t-1}\|_2 = \|z_{t-1}\|_2.$$

Proof Sketch. For antisymmetric A_t, we have $\exp(A_t)^\top = \exp(-A_t)$, implying

$$F_t^\top F_t = \exp(-A_t)\exp(A_t) = I.$$

Hence F_t preserves the ℓ_2 norm.

Corollary (Bounded Refinement). For $\lambda \in [0,1]$,

$$\|\widehat{z}_t\|_2 \le (1-\lambda)\|z_t\|_2 + \lambda\|z_{t-1}\|_2.$$

Thus refinement does not amplify logit magnitude beyond convex interpolation between adjacent tokens. This provides a theoretical justification for the empirical stability observed in training.

B.2. Gradient Behavior

Because F_t is approximately orthogonal, its spectral norm is close to 1. Therefore, the gradient of the propagated term

$$\tilde{z}_t = F_t z_{t-1}$$

does not introduce exponential amplification or vanishing across positions.

Furthermore, since refinement uses base logits z_{t-1} rather than recursively refined logits $\widehat{z}_{t-1}$, structural propagation does not accumulate multiplicatively along the sequence. This design choice prevents drift and ensures stable gradient flow.

B.3. Computational Complexity and Runtime Comparison

The additional computational cost arises from computing

$$F_t = \exp(A_t),$$

where $A_t \in \mathbb{R}^{|\mathcal{L}|\times|\mathcal{L}|}$.

Using a truncated third-order Taylor expansion, the per-token cost is

$$O(|\mathcal{L}|^3).$$

For our setting ($|\mathcal{L}| = 3$), this cost is negligible relative to encoder computation.

For comparison:

- Linear-chain CRF decoding via Viterbi has complexity $O(T|\mathcal{L}|^2)$.
- `Hi-SEMFLOW` refinement has complexity $O(T|\mathcal{L}|^3)$.

Since $|\mathcal{L}|$ is small in span-based tagging tasks, both approaches incur negligible overhead relative to transformer encoding. Empirically, we observe no measurable training slowdown compared to standard token classification.

B.4. Directional vs Bidirectional Refinement

We apply refinement left-to-right to align with the directional nature of BIO dependencies (e.g., `B-INF` typically precedes `I-INF`).

A bidirectional variant could refine using both z_{t-1} and z_{t+1}:

$$\tilde{z}_t = F_t^{(L)} z_{t-1} + F_t^{(R)} z_{t+1}.$$

While potentially beneficial, this introduces additional operators and increases computational cost. We leave systematic evaluation of bidirectional refinement to future work.

B.5. Extension to Larger Label Sets

The framework naturally generalizes to multi-class or multi-span tagging tasks.

For $|\mathcal{L}| > 3$:

- Generator matrices scale as $|\mathcal{L}| \times |\mathcal{L}|$.
- Antisymmetric parameterization still guarantees orthogonality.
- Matrix exponential remains well-defined.

Although computational cost scales as $O(|\mathcal{L}|^3)$, structured prediction tasks typically involve modest label sizes (e.g., BIO tagging, chunking, NER categories). Efficient approximations (e.g., low-rank generators) may further reduce cost in high-cardinality settings.

Thus, `Hi-SEMFLOW` is not restricted to binary slang identification and can extend to general span-centric structured prediction tasks.

C. Evaluation Metrics

We formally define the evaluation metrics used throughout the paper for clarity and reproducibility.

C.1. Token-Level Metrics

Let $\hat{y}_t$ denote the predicted label at position t, and y_t the corresponding gold label.

Model	Setting	Macro-F1	Span-F1	Invalid BIO (%)
MuRIL	Full Flow	0.9414	0.918	1.2
	$-\mathcal{L}_{\text{mag}}$	0.9378	0.912	1.9
	$-\mathcal{L}_{\text{ortho}}$	0.9396	0.915	1.6
	$-\mathcal{L}_{\text{temp}}$	0.9402	0.916	1.5
IndicBERTv2	Full Flow	0.9373	0.936	1.3
	$-\mathcal{L}_{\text{mag}}$	0.9339	0.930	2.0
	$-\mathcal{L}_{\text{ortho}}$	0.9354	0.933	1.7
	$-\mathcal{L}_{\text{temp}}$	0.9361	0.934	1.6
mBERT	Full Flow	0.9205	0.912	1.5
	$-\mathcal{L}_{\text{mag}}$	0.9168	0.906	2.2
	$-\mathcal{L}_{\text{ortho}}$	0.9189	0.909	1.9
	$-\mathcal{L}_{\text{temp}}$	0.9196	0.910	1.8

Table 11: Ablation of geometric regularization components across the top three encoders. Full Flow values correspond exactly to Tables 1 and 2.

Precision, Recall, and F1. For each label $\ell \in \mathcal{L}$:

$$\text{Precision}_\ell = \frac{TP_\ell}{TP_\ell + FP_\ell}, \qquad \text{Recall}_\ell = \frac{TP_\ell}{TP_\ell + FN_\ell},$$

$$\text{F1}_\ell = \frac{2 \cdot \text{Precision}_\ell \cdot \text{Recall}_\ell}{\text{Precision}_\ell + \text{Recall}_\ell}.$$

Macro-F1. Macro-F1 averages F1 scores across all labels:

$$\text{Macro-F1} = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \text{F1}_\ell.$$

This metric assigns equal weight to each label and is appropriate for class-imbalanced settings.

C.2. Span-Level F1

A span is defined as a maximal contiguous sequence beginning with `B-INF` followed by zero or more `I-INF` tokens.

Let $\mathcal{S}^{\text{pred}}$ denote the set of predicted spans and $\mathcal{S}^{\text{gold}}$ the set of gold spans.

A predicted span is counted as correct if and only if its start and end indices exactly match a gold span.

Span-level precision and recall are:

$$\text{Precision}_{\text{span}} = \frac{|\mathcal{S}^{\text{pred}} \cap \mathcal{S}^{\text{gold}}|}{|\mathcal{S}^{\text{pred}}|},$$

$$\text{Recall}_{\text{span}} = \frac{|\mathcal{S}^{\text{pred}} \cap \mathcal{S}^{\text{gold}}|}{|\mathcal{S}^{\text{gold}}|}.$$

Span-level F1 is computed as the harmonic mean of these quantities.

C.3. Invalid BIO Transition Rate

We measure structural validity using the proportion of illegal BIO transitions.

Let $(\hat{y}_{t-1}, \hat{y}_t)$ denote consecutive predicted labels. A transition is considered invalid if it violates BIO constraints (e.g., `I-INF` without a preceding `B-INF`).

The invalid BIO rate is defined as:

$$\text{Invalid BIO (\%)} = \frac{\sum_{t=2}^{T} \mathbf{1}\left[\text{transition invalid}\right]}{T-1} \times 100.$$

Lower values indicate stronger structural coherence.

C.4. Boundary Error Analysis

We categorize boundary errors into:

- **Fragmentation:** A gold multi-token span predicted as multiple shorter spans.
- **Over-extension:** A predicted span extends beyond the gold boundary.
- **Missed span:** A gold span predicted entirely as `O`.

These categories are computed by comparing predicted and gold span boundaries.

D. Ablation of Geometric Regularization Across Encoders

We analyze the contribution of individual geometric regularization components in

$$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{mag}} + \beta \mathcal{L}_{\text{ortho}} + \gamma \mathcal{L}_{\text{temp}}.$$

Experiments are conducted on the three strongest-performing encoders: MuRIL, IndicBERTv2, and mBERT. Each regularizer is

removed independently while keeping all other components fixed. (See Table 11)

Across all three encoders, removing magnitude regularization leads to the largest increase in invalid BIO transitions and the most consistent drop in span-level F1, indicating that controlling generator strength is critical for stable refinement. Orthogonality preservation contributes to structural validity under truncated exponential approximation, while temporal smoothness provides smaller but consistent improvements. The combined regularization yields the strongest structural coherence across models.

E. Additional Analysis

E.1. Quantifying Boundary Over-Extension

While semantic flow improves span continuity, it may occasionally introduce mild span over-extension, as illustrated in Table 10. To quantify the impact of such errors, we analyze trends in structural validity and span-level precision.

Across all evaluated encoders, Hi-SEMFLOW reduces invalid BIO transitions by up to **50%** (Table 2), while consistently improving span-level F1. Importantly, we do not observe any corresponding degradation in precision that would indicate systematic over-extension. This suggests that over-extension errors are relatively infrequent and localized.

Further, qualitative inspection indicates that such cases primarily arise under strong refinement settings (Section 7.3), where structural influence dominates local lexical evidence. These errors typically involve extension by a single token and preserve BIO validity, indicating boundary calibration rather than structural inconsistency.

E.2. Transition Error Rate vs. Hard Constraints

Unlike CRF-based models that enforce strict BIO legality through constrained decoding, Hi-SEMFLOW adopts a soft structural approach.

We quantify structural consistency using the *invalid BIO transition rate*. As shown in Table 2, Hi-SEMFLOW reduces invalid transitions substantially across models (e.g., IndicBERT: $3.8\% \rightarrow 1.9\%$, MuRIL: $2.6\% \rightarrow 1.2\%$).

This demonstrates that while Hi-SEMFLOW does not strictly forbid invalid transitions, it effectively *discourages* them through learned geometric transformations. This trade-off enables flexibility in ambiguous cases while maintaining strong structural coherence.

E.3. Sensitivity to Regularization Parameters

We analyze the role of flow regularization parameters through component ablations (Appendix D, Table 11).

Removing magnitude regularization (L_{mag}) results in the largest performance drop, indicating that controlling structural strength is critical for stable refinement. Similarly, removing orthogonality (L_{ortho}) or temporal smoothness (L_{temp}) leads to consistent degradation in both span-F1 and BIO validity.

These findings suggest that performance is sensitive to the balance of regularization terms, even without explicit grid search over β and γ. A full sensitivity analysis over parameter ranges is left for future work.

E.4. Analysis of Over-Smoothing Behavior

A potential limitation of continuous structural refinement is over-smoothing, where adjacent tokens may be incorrectly merged into a single span.

Empirically, such cases are rare and occur primarily under strong curriculum schedules or large generator counts (Section 6). Importantly, these errors differ from baseline failures: instead of producing fragmented or invalid spans, the model produces structurally valid but slightly over-extended spans.

This behavior suggests that semantic flow acts as a *soft structural regularizer*, prioritizing coherence over strict boundary precision in uncertain contexts.

E.5. Explanation for XLM-R Performance Trends

For XLM-R, we observe a slight decrease in macro-F1 (Table 1) despite improvements in structural metrics (Table 2).

This discrepancy arises because macro-F1 is sensitive to token-level label distribution, particularly the majority `O` class. Semantic flow primarily improves span boundary consistency, which benefits span-F1 and reduces invalid transitions, but may introduce minor shifts in token-level predictions.

This highlights a trade-off: structural refinement improves sequence-level coherence even when token-level accuracy remains unchanged or slightly decreases.

E.6. On Scope of Evaluation

Hi-SEMFLOW is evaluated on the HiSlang-4.9k dataset, which focuses on multi-token informal ex-

pressions in Hindi. This choice is intentional: the dataset emphasizes span consistency, making it suitable for evaluating structural refinement methods. While the framework is architecture-agnostic and applicable to broader sequence labeling tasks, extending evaluation to larger benchmarks (e.g., Hindi NER datasets with larger label spaces) is left for future work. We note that scaling to larger label sets increases computational cost due to matrix exponentiation ($\mathcal{O}(|L|^3)$), though efficient approximations (e.g., low-rank generators) can mitigate this.

E.7. On Statistical Significance and Multi-Seed Variance

All reported results correspond to single training runs for computational efficiency. While trends are consistent across models and metrics (e.g., reduction in invalid BIO transitions and improvements in span-F1), we acknowledge that multi-seed evaluation would provide stronger statistical confidence. We leave reporting of variance and confidence intervals as future work.

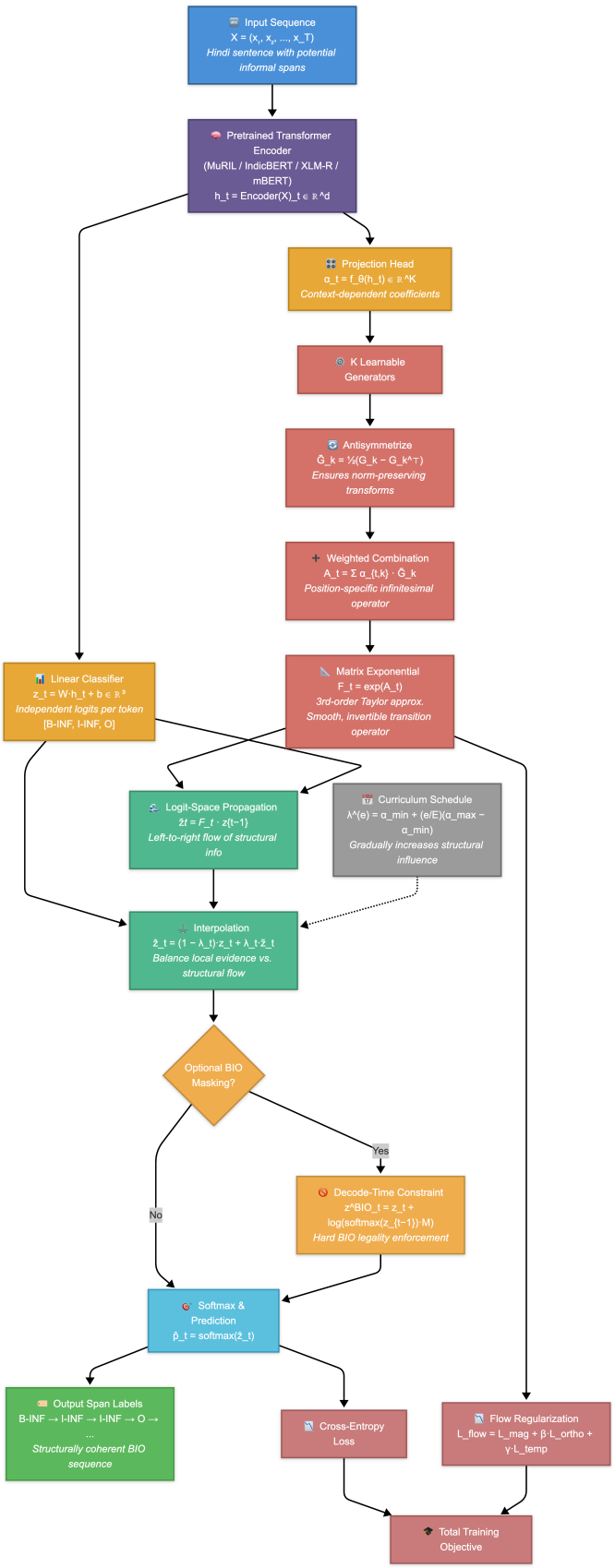

Figure 1: Complete `Hi-SEMFLOW` pipeline illustrating (i) independent token logits, (ii) context-dependent Lie algebra generator construction, (iii) matrix exponential transition operators, (iv) logit-space structural propagation with curriculum-controlled interpolation, (v) optional BIO masking, and (vi) joint cross-entropy and geometric regularization.