Architectural overview—

Core Innovation: Tri-Modal Adversarial Transparency Network (TATN)

Revolutionary Architecture Components

1. Dynamic Span-Sense Co-Detection Module (DSCD)

- **Novel Innovation**: Unlike traditional two-stage approaches, this module simultaneously learns span detection and sense disambiguation in a shared embedding space using **contrastive multi-task learning**
- **Unique Feature**: Employs **uncertainty-aware attention** that dynamically adjusts focus based on contextual ambiguity levels
- **Technical Breakthrough**: Uses **semantic density clustering** to identify homograph candidates without predefined sense inventories

2. Adversarial Sense Balance Network (ASBN)

- **Revolutionary Approach**: Implements **hierarchical adversarial training** with three discriminators:
    - Frequent-sense discriminator (combats bias toward common meanings)
    - Context-sparse discriminator (handles minimal context scenarios)
    - Cross-lingual sense discriminator (ensures translation consistency)
- **Novel Technique**: **Gradient reversal with confidence weighting** prevents over-reliance on statistical artifacts

3. Transparent Rationale Generator (TRG)

- **Unprecedented Feature**: Generates human-interpretable explanations for disambiguation decisions in real-time
- **Innovation**: Uses **attention-based evidence extraction** combined with **natural language rationale synthesis**
- **Unique Capability**: Provides confidence scores and alternative sense rankings with linguistic justification

A. Homograph detection and disambiguation :
Dynamic Span-Sense Co-Detection Module (DSCD)

- **Novel Innovation**: Unlike traditional two-stage approaches, this module simultaneously learns span detection and sense disambiguation in a shared embedding space.
- **Unique Feature**: Employs **uncertainty-aware attention** that dynamically adjusts focus based on contextual ambiguity levels
- **Technical Breakthrough**: Uses **semantic density clustering** to identify homograph candidates without predefined sense inventories

Dynamic Span-Sense Co-Detection Module (DSCD)

Goal:- Jointly detect spans of homographs and disambiguate their senses in real-time using a single neural model.

1. **Sentence Tokenization**
2. **Normalization & Cleaning**
3. **Subword Segmentation (SentencePiece)**
4. **Build Contextual Embedding**
5. **Uncertainty-Aware Attention-**
    a. focus the model's attention on potentially ambiguous words (homographs) using uncertainty.
    b. For each embedding, **predict its probability distribution** over possible word translations/senses (eg.- softmax output).
    c. **Calculate uncertainty score** for each token via entropy. The attention mechanism internally can **focus more on tokens with higher entropy** to guide the span detector head in identifying ambiguous words.
    Low Uncertainty means the model is confident: likely unambiguous.
    d. **Dynamic Span Detection:** Set a threshold value for uncertainty.
    If **Uncertainty value> threshold** value then it is flagged as an ambiguous word in that context.
    e. **Attention Adjustment:** The encoder can give greater *attention weight* to tokens where Uncertinty is high, meaning that potentially ambiguous tokens exert more influence during context modeling and decision making.
6. **Semantic Clustering (Unsupervised):** During model development, cluster contextual embeddings of all occurrences of high-uncertainty tokens into K clusters, means :-

    For each ambiguous word in the vocabulary (e.g., "পাতা" = "leaf/page/blade"):
    - Collect all contextual embeddings from various contexts in the corpus.
    - Cluster embeddings (e.g., using KMeans) into *K* sense clusters (e.g., cluster 0: "leaf", cluster 1: "page", cluster 2: "blade").
    - Save cluster centroids and assign cluster IDs as pseudo-gold sense labels for these word instances.

7. **Span and Sense Multi-Task Prediction :**
    For each token it formulates the joint prediction of-
    a. **Span Detector Head:** For each token , predict a binary label $\{0,1\}$ indicating whether that token is ambiguous (homograph) in the sentence (1 = ambiguous, 0 = unambiguous).
    b. **Sense Classifier Head:** Predicts sense cluster ID for ambiguous tokens. For tokens flagged ambiguous, predict its *numeric* sense cluster $\in\{1,\ldots,K\}$ where K is the number of discovered sense clusters (e.g., 3 for "পাতা")

    Both heads share the **same contextual embedding representation** of tokens, enabling efficient, joint learning.

8. **Sense-Augmented Representation for Translation:**
   Enrich Token Representation :
   a. For flagged ambiguous tokens, the model adds the embedding of the assigned sense cluster to the token embedding.
   b. This creates a sense-aware contextual vector for use by the translation decoder.

9. **Sense-Aware Decoding :** The decoder receives all (sense-enhanced) token embeddings. And based on it ,the model translate the sentence from Bengali to English.

B. Adversarial Sense Balance Network (ASBN)

Approach: Implements hierarchical adversarial training with three discriminators:

- **Frequent-sense discriminator** (combats bias toward common meanings): Detects when the encoder's sense predictions follow frequency statistics (e.g., always picking the most common sense). It forces the encoder to "hide" sense-frequency patterns, encouraging true context-driven predictions.

Example

Let's say the Bengali word **"পাতা"** most often means "leaf" in training data, but can also mean "page" (of a book), "blade" (of a knife), or "card" (as in playing card):

- **Sentence 1:** "গাছের পাতা সবুজ।" ("The leaf of the tree is green.")
  — Context really does demand "leaf".
- **Sentence 2:** "সে পাতা দিল।" ("He gave the পাতার." — ambiguous; could be "card", "leaf", "page")
- **Sentence 3:** "বইয়ের পাতা ছেঁড়া।" ("The page of the book is torn.")
  — Context demands "page", but "leaf" is the most common overall.

What the Discriminator Does

- It receives the sense prediction ("পাতা" = "leaf").
- It checks if this matches the most frequent overall sense.
- If it can predict the model's sense choice *without looking at the context* (pure frequency), that reveals the encoder is not using rich context — it's "lazy."
- When this happens, **a gradient reversal penalty is applied**, forcing the encoder (DSCD) to refine its context use so senses are less predictable by frequency alone.

- **Context-sparse discriminator** (handles minimal context scenarios): Detects if the encoder is "guessing" or relying on patterns when the local context is vague or insufficient. Encourages the encoder to attend harder to global or subtle cues—not just generic patterns.

Example

Suppose the sentence is very short or does not contain distinctive context words:

- **Sentence 4:** “পাতা পড়ল।” (“পাতা fell.”)
- This sentence could mean “The leaf fell,” “The card fell,” or even “The page fell,” depending on situation.

What the Discriminator Does

- It identifies such cases of sparse or ambiguous context.
- It analyzes whether DSCD always chooses the most frequent sense (e.g., “leaf”).
- If so, it penalizes the encoder for not expressing enough uncertainty, not looking for global cues, or just exploiting data priors.
- This **forces the model to be cautious, to signal ambiguity, or to search for less obvious contextual evidence**.

- **Cross-lingual sense discriminator** (ensures translation consistency): Detects mismatches between source and target sense choices (e.g., translating to a less-specific or wrong sense in the target language). It keeps the translated sense aligned, especially when ambiguous words have different sense frequencies across languages.

Example

- **Sentence 5:** “বইয়ের পাতা ছেঁড়া।”
  Suppose the encoder’s sense prediction for "পাতা" is “leaf” (wrong), but the target translation should be “page.”
- Or, similarly:
- **Sentence 6:** “ছুরির পাতা ধারালো।” (“The blade of the knife is sharp.”)
  The model predicts “leaf” (wrong), but correct would have been “blade”.

What the Discriminator Does

- It compares the predicted Bengali sense for "পাতা" with the semantic role of the word in the English sentence.
- If it sees **systematic mismatches**, it adversarially penalizes the encoder, **forcing future representations to better align semantic senses across source and target**.
- This reduces translation errors caused by sense collapse or divergence.

ASBN(Adversarial Sense Balance Network) architectural Overview:-

1. **Model Foundation :-** DSCD Module: Predicts, for each potentially ambiguous token, a sense cluster among K choices using context and clustering, and detects ambiguous spans.
2. **ASBN:** Frequent-Sense Discriminator, Context-Sparse Discriminator , Cross-Lingual Sense Alignment Discriminator all these act as “bias detectors”—each trying to spot a systematic error.
   Steps:

A. Frequent-Sense Discriminator(D_{freq})

For each ambiguous token :

b. Get the sense prediction(one of K clusters).
c. The discriminator D_{freq} gets as input : The context embedding & The sense assignment.

D_{freq} tries to predict whether the most frequence is the *most frequent* sense (binary label: 1 for most frequent, 0 otherwise).

B. Context-Sparse Discriminator (D_{ctx})

Detects if the DSCD is ignoring weak context (making a guess where not enough information is present).

It takes input: The context length for the ambiguous word & The token embedding and sense assignment.

Predicts if the context is "sparse" (binary label: 1 for sparse, 0 for rich) and if sense assignment matches sparse-context frequency.

C. Cross-Lingual Sense Alignment Discriminator (D_{xl})

Checks if the predicted Bengali sense maps correctly to the English translation's sense, enforcing cross-lingual consistency.

For each ambiguous source token w:

Identify predicted sense, Extract the aligned English token's embedding.

D_{xl} tries to predict if the target translation corresponds to the *same* sense cluster.

D. Confidence-Weighted Gradient Reversal

Compute confidence for each sense assignment (e.g., softmax probability pmax from the sense head).

Scale each adversarial loss by model confidence:

More confident, more adversarial penalty. Less confident (uncertain), less penalty.

E. Training Objective and Optimization

All networks are updated jointly using backpropagation, with special gradient reversal in the adversarial branches.

TRG: Transparent Rationale Generator

TRG makes model's disambiguation and translation decisions truly **interpretable and transparent** by automatically generating human-readable explanations—claiming, evidencing, and justifying every sense decision made by DSCD (and improved by ASBN). The result is a translation system that not only outputs the answer, but also tells in clear language why it made each ambiguous word choice.

1. Joint Explanation Extraction and Synthesis (Rationale-in-the-Loop)

While most explanation models merely show token-level attention heatmaps or post-hoc templates, TRG in TATN dynamically and *jointly*:

- **Extracts salient context tokens** (words or phrases most influential for a sense decision), using both attention weights and uncertainty/entropy scores from DSCD.
- **Synthesizes natural-language rationales** via a lightweight, context-conditioned sequence generator.

How It's Different and Novel

- **Not just showing attention weights.** TRG produces fluent explanations, not just highlight overlays.
- **Conditioned on sense decisions AND adversarial context.** The rationale generation is guided by not just "why this sense," but also "why not the others"—with uncertainty and ASBN audit scores as explicit context for the generation.
- **Handles rare, overlapping, or unseen senses.** It can *explain away* why a non-frequent or rare cluster is chosen, using context tokens and even mentioning ASBN "criticisms."

Concrete Example: Bengali "মূল" ("mool")

Ambiguous word: **"মূল"** (possible senses: root (plant), price, radical/square root, origin/source).

Input sentence:
"৯ এর বর্গমূল হল ৩।"
("The square root of 9 is 3.")

Step-by-Step Novel TRG Process

1. Collect Sense Decision & Contextual Cues

- DSCD predicts for "মূল" in this context:
    - Sense cluster: **radical/math root** (from "বর্গমূল", "৯", "৩")
    - Model's confidence: e.g., 95%
    - Salient context tokens by attention: "বর্গ" (square), numerals
    - (ASBN during training): minimal bias detected, strong context

2. Prepare Explanation Inputs

- Chosen sense and alternatives:
    - Chosen: radical/math root
    - Alternatives: root (plant), price, origin
- Salient context: "বর্গ", "৯", "৩"
- Reasoning for exclusion: No plant/sale/foundation words; mathematical terms dominate.

3. Generate Natural Language Rationale

- TRG's sequence generator (small, specialized neural module) uses:
    - What context tokens were crucial ("square," "9," "3")
    - Which sense was chosen, and confidence
    - Why other senses were not chosen (no plant or price words in this context)
    - Optionally, any detected uncertainty or adversarial audit remarks

Step by step implementation of TRG module:

Word Example: "মূল" (*mool*)

- **Senses**:
    1. Root (of a plant)
    2. Price (of a thing)
    3. Radical/square root (math)
    4. Origin/source/foundation

Sample Sentence for Disambiguation:

"এই বইয়ের মূল কত?"
("What is the price of this book?")

1. Collect Model Decisions and Internal Signals

a) Get DSCD Outputs for "মূল"

- **Sense prediction:** Price (from context "বইয়ের"/"of the book" and "কত"/"how much")
- **Sense probabilities (softmax):** [0.08 (root), **0.86 (price)**, 0.03 (radical), 0.03 (origin)]
- **Model confidence:** 86% for "price"
- **Ambiguity flagged:** Yes (from DSCD's span head)

b) Context Saliency Extraction

- Use attention weights & context cues:
    - Most salient: “বইয়ের” (of the book), “কত” (how much), presence of question structure
    - No plant, math, or origin/foundation words

c) (Optional, Training Only) ASBN Audit Info

- ASBN confirms: Not defaulting to most frequent sense ("root"); choosing “price” because context cues are strong

2. Build Structured Input for the Rationale Generator

- **Inputs to TRG:**
    - Token: “মূল”
    - Chosen sense: Price
    - Top context clues: “বইয়ের”, “কত”
    - Sense probabilities & confidence
    - Alternative senses not chosen (and reason: missing context cues)
    - ASBN bias score (if relevant) and uncertainty

3. Generate Fluent Explanation (Rationale Generator Module)

- **Model:**
  Use a lightweight sequence-to-sequence generator (small Transformer, GRU, or template-augmented model) specially trained/tuned to verbalize:
    - Why this sense was picked ("price")
    - Evidence from context (words, question form)
    - Why other senses were rejected (e.g., “No plant/math/origin context words nearby”)
    - Model’s certainty
    - (If relevant) "Critiqued" decisions (e.g., "Most frequent sense not chosen thanks to supporting evidence")