How would joint training influence the performance of a pipeline consisting of Encoder and Neural Network?

As pertains to my research question:

Forcing the encoder output itself to satisfy geometry degrades representations.

Goal of my research question:

The goal is to test/create a system that:

- creates **valid analogies between fields**
- works **zero-shot for at least one domain**
- relies on a **latent space where the main operations occur**

Core architectural components identified

1. **Encoder**
    - Always processes all inputs (including new-domain text)
    - Defines the latent coordinate system
    - Does *not* merely "embed" text, but must learn a geometry suitable for analogy
2. **Latent-space operations (analogy mechanism)**
    - Perform structured transformations (relations, mappings, role substitutions)
    - Assume certain geometric regularities (linearity, composability, alignment)
3. **Optional upstream selector**
    - Narrows which objects or relations are considered
    - Reduces combinatorial burden
    - Must operate in the same latent space

Key conclusions about training for my pipeline

Freezing the encoder is incompatible with your goal

- Freezing does **not** bypass the encoder; new inputs still go through it
- But freezing prevents the encoder from adapting its geometry

- This causes misalignment between:
    - where inputs are placed
    - and what downstream analogy operations assume

Conclusion:
If latent geometry matters operationally, the encoder cannot be frozen.

Is Joint training is structurally necessary for my purposes?

Joint training is required because:

- the analogy mechanism defines what “good geometry” means
- only joint optimization lets gradients from analogy failures reshape the encoder
- without it, components live in incompatible coordinate systems

Conclusion:
Joint training is how the encoder internalizes the relational geometry your pipeline relies on.

Zero-shot analogy does not require seeing the new domain

The encoder does *not* need domain-specific data if:

- relations are abstract and reusable
- training forces them to be represented as separable, composable factors
- geometry clusters by **relational role**, not domain identity

Zero-shot works when:

- new-domain structures lie in the span of learned relational primitives

“No truly novel relations” is partly correct but insufficient

- Many domains share abstract relational types
- Analogy is possible because of this
- But ML models only learn relations they are *forced* to represent abstractly

Key refinement:
Novelty is about **coverage of relational bases**, not existence of relations.

Pure text-only neural networks can only approximate this behavior

- Large text-trained models exhibit emergent analogies
- But their latent geometry is:
    - accidental
    - not explicitly constrained
    - unreliable for systematic analogy

Conclusion:
Text-only training can work approximately, but is insufficient if analogy is a core, reliable operation.

Role of an upstream mechanism

- Selects which latent objects/relations are worth comparing
- Reduces burden on the encoder
- Improves precision without breaking zero-shot
- Must be **jointly trained**, not domain-specific

Final Conc.

To achieve reliable zero-shot analogies across domains, the encoder, any relation-selection mechanism, and the analogy operations must be **jointly trained** so that the latent space geometry explicitly supports abstract, composable relational structure. Relying on frozen encoders or purely implicit geometry learned from text alone makes analogy fragile and misaligned with downstream operations.