Fine-tuning a multimodal transformer model on limited GPU resources requires optimization techniques:

1. **Use Mixed Precision Training**: Implementing FP16 (float16) reduces memory usage and speeds up training.

2. **Gradient Accumulation**: Instead of large batch sizes, accumulate gradients over multiple smaller steps.

3. **Efficient Data Loading**: Use `torch.utils.data.DataLoader` with `num_workers` to optimize loading speeds.

4. **Layer Freezing**: Freeze lower layers and fine-tune only the top layers to reduce computational load.

5. **Parameter Efficient Fine-Tuning (PEFT)**: Techniques like LoRA (Low-Rank Adaptation) allow training small parameter subsets.

6. **Cloud or Colab TPUs**: If local GPU is insufficient, leverage free/affordable cloud services like Google Colab or Hugging Face Spaces.

By combining these techniques, you can achieve efficient model training while minimizing hardware constraints.